Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского федерального университета
Дата подписания: 18.04.2024 15:49:04
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

по выполнению лабораторных работ
по дисциплине
«ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ»
для направления подготовки **10.03.01 Информационная безопасность**
направленность (профиль) **Безопасность компьютерных систем**

Пятигорск
2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Цель и задачи изучения дисциплины

Цель дисциплины: формирование набора универсальных и общепрофессиональных компетенций будущего бакалавра (специалиста) по направлению подготовки 10.03.01 Информационная безопасность.

Задачами дисциплины являются: дать основы о методах и средствах защиты информации в компьютерных системах; дать основы правил разграничения доступа и основных функций СЗИ, его обеспечивающих; дать основы практических аспектов построения систем ограничения доступа и других СЗИ; дать основы аппаратной реализации различных средств защиты информации; дать основы о защитных механизмах, реализованных в средствах защиты компьютерных систем от несанкционированного доступа (НСД); дать основы вопросов защиты ПО от несанкционированного использования; дать основы о применении средств криптографической защиты информации и средств защиты от НСД для решения задач обеспечения информационной безопасности; дать основы методов защиты от РПВ; дать основы методов и особенностей защиты объектов ОС; дать основы принципов построения файловой системы и моделей разграничения доступа к объектам.

2. Оборудование и материалы

Для проведения практических занятий необходимо следующее материально-техническое обеспечение: персональный компьютер; проектор; возможность выхода в сеть Интернет для поиска по образовательным сайтам и порталам; интерактивная доска.

3. Наименование лабораторных работ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов
5 семестр		
13	**Лабораторная работа 1.** Программно-аппаратные средства защиты от DDoS атак.	6
13	**Лабораторная работа 2.** Межсетевые экраны.	6
13	**Лабораторная работа 3.** Построение и управление RAID – массивами и логическими томами.	8
13	**Лабораторная работа 4.** Программное восстановление данных.	8
17	**Лабораторная работа 5.** Обнаружение и предотвращение вторжений.	8
	Итого за 5 семестр	**36**
6 семестр		
18	**Лабораторная работа 6** Электронная цифровая подпись.	8
18	**Лабораторная работа 7.** Программно-аппаратное шифрование данных при их хранении	8
19	**Лабораторная работа 8** Защита программ от НСИ с помощью USB-ключей.	8
22	**Лабораторная работа 9.** Sandbox – файловые антивирусы.	8
	Итого за 6 семестр	32

	Итого	**68**

4. Содержание лабораторных работ

ЛАБОРАТОРНЫЕ РАБОТЫ

1.Лабораторная работа №1. Протокол Kerberos.

Цель работы: Получить теоретические и практические навыки защиты от DDoS атак.

1. Теоретическая часть

DoS (от англ. Denial of Service — отказ в обслуживании) — хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т.д.). Но чаще это мера экономического давления: простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

Если атака выполняется одновременно с большого числа компьютеров, говорят о **DDoS-атаке** (от англ. Distributed Denial of Service, распределённая атака типа «отказ в обслуживании»).

«Медленные атаки». Атаки типа SlowLoris. Атаки этого типа открывают множественные соединения к целевому web серверу и держат их открытыми как можно дольше путем посылки незавершенных HTTP запросов. Атакуемый сервер ждет завершения запроса оставляя соединение открытым. Периодически, SlowLoris посылает последующие заголовки HTTP для каждого запроса, но не пытается завершить соединение. В итоге пул подключений сервера заполняется и попытки доступа новых (легитимных) соединений отклоняются. Для успешной реализации атаки необходимо наличие уязвимой для атаки веб-службы (apache или сервис с похожей реализацией многопоточности) и отсутствие альтернативных средств влияющих на успешное проведение атаки (например, балансировщиков нагрузки).

TCP connection flood. Атаки этого типа инициируют большое количество соединений, но не пытаются завершить эти соединения в течение длительного времени, что задействует ресурсы сервера. Одним из подходов к реализации атаки является посылка большого количества SYN запросов, что, похоже, на атаки типа SYN flood, за исключением того, что используются реальные IP адреса и соединение корректно устанавливается. От таких атак можно защититься с помощью установки ограничения на количество соединений с IP адреса.

Однако новые методы атаки – «медленные» используют низкоскоростное соединение, в котором устанавливают соединение с сервером спустя несколько секунд после инициации и поддерживают соединение длительный период времени, регулярно посылая пакеты с данными формально соответствующими протоколу к серверу. Каждый IP адрес с которого производится атака может поддерживать сотни таких соединений, в то время как количество соединений сервера ограничено, что может привести к отказу в обслуживании. В качестве примера, ярким представителем такого класса атак является Slow Read attack.

«Произвольные пакеты» (случайные данные в пакетах по соответствующим протоколам). HTTP Flood Attack. Типичная работа веб браузера заключается в посылке HTTP запросов, в первую очередь GET и POST. Запрос GET служит для получения статичных ресурсов, таких как изображения; запрос POST используется для организации доступа к динамическому содержимому страниц. Атака наиболее эффективна в случаях когда вынуждает сервис выделить максимально доступный ресурс в ответ на простой единичный запрос. Атаки, использующие POST запросы, зачастую более ресурсоемки, инициируют сложную обработку на стороне сервера. С другой стороны атаки использующие запросы GET просты в создании и легко масштабируемы.

Атака случайными DNS-запросами на несуществующие домены. В атаках этого типа атакующий пытается переполнить количество одновременно-возможных DNS запросов, наводняя сервер запросами к множеству несуществующих доменов.

UDP Flood. Атаки этого типа используют особенности UDP протокола, где атакующий может создавать пакеты большого размера произвольного содержимого. Когда атакуемый узел получает такие пакеты на определенный порт, то выполняется проверка приложения, которое должно «слушать» этот порт. Когда выясняется отсутствие ассоциированного приложения, то выполняется посылка ответа в виде ICMP пакета Destination Unreachable. Таким образом, при большом количестве UDP пакетов, атакуемая система будет занята посылкой множества ICMP пакетов. Кроме того, UDP Flood может эффективно применяться для атаки на канал.

«SSL». Атака некорректными SSL/TLS пакетами. Атаки этого посылают произвольные данные к целевому серверу. SSL/TLS протоколы требуют выделения на стороне сервера значимого количества ресурсов, которые затрачиваются на попытки распознать произвольные данные в качестве легитимного запроса на инициацию SSL “рукопожатия”

Атака медленными SSL/TLS пакетами. Атаки этого типа, как и прочие «медленные» требуют небольшого количества пакетов для достижения отказа в обслуживании на относительно большом сервере. Атакующий в этом случае должен инициировать обычное SSL «рукопожатие» затем запросить пересмотр ключа шифрования, периодически повторяя этот ресурсоемкий для стороны сервера запрос, атакующий может исчерпать ресурсы атакуемого узла.

Атака SSL/TLS renegotiation. Атаки этого типа эксплуатируют слабость алгоритма «рукопожатия» SSL/TLS, где потребности в мощности процессора на стороне сервера на порядок превышают таковые на стороне клиента. Таким образом, если вычислительная мощность клиента и сервера равны с точки зрения реализации функций алгоритма RSA, то клиент может подавить сервер, посылая ему множество запросов инициации переговоров.

«Некорректные пакеты/фрагменты». Фрагментированные пакеты (UDP Fragment Flood). Атаки этого типа посылают UDP датаграммы, которые произвольно ссылаются на другие, отсутствующие в потоке датаграммы, что приводит к повышенному потреблению памяти на стороне атакуемого узла.

Некорректные IP фрагменты. Большой класс атак ориентируется на эксплуатацию уязвимостей в поддержке фрагментации пакетов в протоколе IP. Атаки этого типа могут приводить к отказу в обслуживании, или использоваться для обхода функций безопасности системы обнаружения вторжений. Одной из популярных атак этого типа является пересечение IP фрагментов. Атака может быть выполнена в случае, когда два фрагмента в одной и той же IP датаграмме имеют смещения, указывающие что они перекрывают друг друга при позиционировании. Некоторые системы неспособны корректно обработать подобные ситуации.

Неверные значения в заголовках пакетов. Атаки этого типа ориентируются на отдельные операционные системы и приложения, которые ошибочно обрабатывают некорректные значения в пакетах. Таким образом атаки этого типа могут приводить к

отказу в обслуживании, или использоваться для обхода функций безопасности системы обнаружения вторжений. Широко известным примером такого типа атаки может быть Land attack. В которой атакующий использует подмену IP адреса в заголовке пакета в комбинации с созданием TCP соединения. Причем устанавливается один и тот же адрес для источника и приемника, что приводит к попыткам установить соединение с самим собой.

«SMTP». SMTP Flood. В атаках этого типа злоумышленник пытается установить соединение с почтовым сервером и отправляет произвольных писем на сгенерированные случайным образом адреса, либо бездействует до истечения таймаута, удерживая соединение открытым. Каждое SMTP соединение утилизирует часть ресурсов сервера, тем самым атакующий пытается вызвать отказ в обслуживании.

1.2.Программы для выполнения лабораторной работы

hping3:

Установка:

```
sudo apt-get install hping3
```

Использование:

```
sudo hping3 -i u1 -S -p 80 192.168.1.1
```

где S - SYN, p 80 - порт 80, i u1 - Ждать 1 микро секунду между каждым пакетом

Можно также указать количество пакетов параметром - c :

```
sudo hping3 -i u1 -S -p 80 -c 10  192.168.1.1
```

slowhttptest:

Установка:

Скачать последнюю версию можно здесь:

https://code.google.com/p/slowhttptest/downloads/list

```
tar -xzvf slowhttptest-x.x.tar.gz
cd slowhttptest-x.x
./configure --prefix=PREFIX
make
sudo make install
```

Здесь PREFIX должен быть заменён на абсолютный путь, где инструмент slowhttptest должен быть установлен.

У вас должна быть установлена libssl-dev <sudo apt-get install libssl-dev> для успешной компиляции этого инструмента. Большинство систем должны иметь его.

Использование. Режим SlowLoris:

```
slowhttptest -c 1000 -H -i 10 -r 200 -t GET -u http://192.168.1.37/info.php -x 24 -p 3
```

THC-SSL-DOS:

Установка:

```
sudo apt-get install libssl-dev
```

Скачивайте отсюда http://www.thc.org/thc-ssl-dos/

Разархивируйте, переходите в папку, выполните:

```
./configure
sudo make all install
```

Использование:

```
thc-ssl-dos 192.168.1.202 --accept
```

ddosim:

Установка:

sudo apt-get install libcap-dev
sudo apt-get install build-essential
Скачать ddosim можно здесь:
http://sourceforge.net/projects/ddosim/
Разархивируйте, перейдите в папку, выполните:
sudo ./configure
sudo make
sudo make install
Использование:
invalid HTTP requests:
./ddosim -d 192.168.1.2 -p 80 -c 10 -r HTTP_INVALID -i eth0
TCP connection flood:
./ddosim -d 192.168.1.2 -p 80 -c 0 -i eth0
HTTP valid requests:
./ddosim -d 192.168.1.2 -p 80 -c 0 -w 0 -t 10 -r HTTP_VALID -i eth0
SMTP:
./ddosim -d 192.168.1.2 -p 25 -k 10.4.4.0 -c 0 -r SMTP_EHLO -i eth0

GoldenEye:
Установка:
mkdir goldeneye
cd GoldenEye
wget https://github.com/jseidl/GoldenEye/archive/master.zip
unzip master.zip
cd GoldenEye-master/
Использование:
./goldeneye.py <URl>

Http Unbearable Load King:
Скачать можно здесь:
https://packetstormsecurity.com/files/112856/HULK-Http-Unbearable-Load-King.html
Запуск
python ./hulk.py 192.168.1.113 safe

Fudp - UDP flood:
Установка:
Скачать можно здесь
http://www.olszewski.cc/fudp/
Переходите в папку, которую только что извлекли из архива и выпоняете <sudo make>
Использование.
Команда для создания вывода:
tcpdump -qt -i lo -n proto UDP
Запуск командой:
./fudp -l *цель*

1.3.Защита от DDoS

Защита от SYN-флуда
Защита строится на отключении очереди «полуоткрытых» TCP-соединений:

- sudo sysctl -w net.ipv4.tcp_max_syn_backlog=1024

Включение механизма TCP syncookies:

- sudo sysctl -w net.ipv4.tcp_syncookies=1

Ограничение максимального числа «полуоткрытых» соединений с одного IP к конкретному порту:

- sudo iptables -I INPUT -p tcp --syn --dport 80 -m iplimit --iplimit-above 10 -j DROP

Защита от UDP-флуда:

- Так как UDP-пакеты отсылаются на различные UDP-сервисы, то достаточно просто отключить их от внешнего мира и установить ограничение на количество соединений к DNS-серверу:

- sudo iptables -I INPUT -p udp --dport 53 -j DROP -m iplimit --iplimit-above 1

Защита от ICMP-флуда:

• Для того, чтобы защититься от ICMP-флуда нужно отключить ответы на запросы ICMP ECHO:

- sudo sysctl net.ipv4.icmp_echo_ignore_all=1

или с помощью брандмауэра:

- sudo iptables -A INPUT -p icmp -j DROP --icmp-type 8

Действия в начале DoS:

- Для определения SYN-флуда необходимо установить число «полуоткрытых» соединений:

- grep ":80\ " | grep SYN_RCVD

В идеале их не должно быть совсем (максимум 1-3). Если это не так, то можно говорить о наличии DoS-атаки. Сложнее обстоит дело с HTTP-флудом. В самом начале необходимо подсчитать количество подключений на 80 порт и количество процессов Apache. Если они значительно превышают среднестатистические, то следует задуматься.

- grep httpd | wc -l grep ":80\ " | wc -l

Список IP-адресов, с которых идут запросы на подключение можно посмотреть следующей командой:

- grep ":80\ " | sort | uniq -c | sort -nr | less

Далее следует провести анализ пакетов с помощью команды tcpdump:

- sudo tcpdump -n -i eth0 -s 0 -w output.txt dst port 80 and host IP-сервера

Если наблюдается большой поток однообразных пакетов с разных IP-адресов, которые направлены на один порт, можно смело предполагать, что имеет место DoS-атака. Далее необходимо сбросить все эти соединения:

- sudo iptables -A INPUT -s xxx.xxx.xxx.xxx-p tcp – destination-port http -j DROP

2.3.Задания к лабораторной работе

• Для выполнения данной лабораторной работы нужны две виртуальные машины, 1 - с установленным web-сервером, и 2 - которая будет проводить DoS-атаку. На вторую тоже нужно поставить web-сервер, так как она тоже может быть целью некоторых программ (slowhttptest), выполняющих DoS атаку.

• Произведите DDoS атаки следующих типов (В теоретической части есть описание того, как устанавливать и использовать каждую из программ для проведения атак):

1. SYN flood используйте hping3
2. SlowLoris используйте slowhttptest
3. SSL используйте THC-SSL-DOS
4. TCP connection flood используйте ddosim
5. HTTP DDoS with valid requests используйте ddosim
6. HTTP DDoS with invalid requests используйте ddosim
7. Для HTTP flood используйте Goldeneye или Http Unbearable Load King
8. Для UDP flood используйте Fudp

• Защитите сервер от DDoS.

1.4. Вопросы к лабораторной работе

1. В чем отличие DDoS от DoS?
2. Что такое UDP-флуд?
3. Как защититься от DDoS?
4. Что такое SYN-флуд?
5. Как определить что используют SYN-флуд?
6. Как защититься от HTTP-флуда?
7. Как работают атаки типа SlowLoris?
8. Как совершить DDoS-атаку на почтовый сервер?
9. Как работают атаки типа SSL?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

2.Лабораторная работа №2. Межсетевые экраны (Iptables, WEB APLICATION FIREWALL).

Цель работы: Изучение межсетевых экранов. Приобретение навыков работы с Iptables и WAF.

1.1. Теоретическая часть.

Межсетевой экран

Скорее всего, ранее вы уже сталкивались с таким понятием как межсетевой экран. В ядро Linux встроен свой межсетевой экран, называемый Netfilter. Управление им осуществляется с помощью утилиты Iptables.

Межсетевой экран, сетевой экран, файервол, брандмауэр — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами. Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.

Рассмотрим принцип работы Netfilter. Когда сетевые пакеты попадают в сетевой интерфейс, они после ряда проверок ядром проходят последовательность так называемых цепочек. Пакет обязательно проходит через цепочку PREROUTING, после чего определяется, кому он, собственно, был адресован. Если пакет не адресован локальной системе (в нашем случае серверу), он попадает в цепочка FORWARD, а иначе — в цепочку INPUT, после прохождения которой отдается локальным демонам или процессам. После этого при необходимости формируется ответ, который направляется в цепочку OUTPUT. После цепочек OUTPUT или FORWARD пакет в очередной раз встречается с правилами маршрутизации и направляется в цепочку POSTROUTING. В результате прохождения пакетом цепочек фильтрации несколько раз, проверка его принадлежности определенным критериям осуществляется несколько раз. В соответствии с этими проверками к пакету применяется определенное действие:

- ACCEPT — пакет «принимается» и передается в следующую цепочку.
- DROP — удовлетворяющий условию пакет отбрасывается и не передается в другие таблицы или цепочки.
- REJECT — пакет отбрасывается, но при этом отправителю отправляется ICMP-сообщение, сообщающее об отказе.
- RETURN — пакет возвращается в предыдущую цепочку и продолжает её прохождение начиная со следующего правила
- SNAT — применить трансляцию источника в пакете. Используется только в цепочках POSTROUTING и OUTPUT таблицы nat.
- DNAT — применить трансляцию адреса назначения в пакете. Используется в цепочках PREROUTING и (очень редко) OUTPUT в таблице nat.

Основные команды Iptables

Параметр	Описание	Пример
–destination(-d)	IP адрес назначения пакета. Может быть определен несколькими путями (см. –source).	iptables -A INPUT –destination 192.168.1.0/24
–in-interface (-i)	Определяет интерфейс, на который прибыл пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках INPUT,FORWARD и PREROUTING. Возможно использование знака +, тогда подразумевается использование всех интерфейсов, начинающихся на имя+ (например eth+ - все интерфейсы eth).	iptables -t nat -A PREROUTING –in-interface eth0
–out-interface(-o)	Определяет интерфейс, с которого уйдет пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках OUTPUT, FORWARD и POSTROUTING. Возможно использование знака +.	iptables -t nat -A POSTROUTING –in-interface eth1
	Неявные (необщие) параметры	
-p proto -h	Вывод справки по неявным параметрам протокола proto.	iptables -p icmp -h
–source-port(–sport)	Порт источник, возможно только для протоколов –protocol tcp, или –protocol udp	iptables -A INPUT –protocol tcp –source-port 25
–destination-port(–dport)	Порт назначения, возможно только для протоколов –protocol tcp, или –protemocol udp	iptables -A INPUT –protocol udp –destination-port 67
	Явные параметры	
-m state –state (устарел)	Состояние соединения. Доступные опции: NEW (Все пакеты устанавливающие новое соединение) ESTABLISHED (Все пакеты, принадлежащие установленному соединению) RELATED (Пакеты, не принадлежащие установленному	iptables -A INPUT -m state –state NEW, ESTABLISHED iptables -A INPUT -

Параметр	Описание	Пример
он же -m conntrack –ctstate	соединению, но связанные с ним. Например - FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связаны.) INVALID (Пакеты, которые не могут быть по тем или иным причинам идентифицированы).	m conntrack –ctstate NEW, ESTABLISHED
-m mac –mac-source	Задает MAC адрес сетевого узла, передавшего пакет. MAC адрес должен указываться в форме XX:XX:XX:XX:XX:XX.	-m mac –mac-source 00:00:00:00:00:0
	Дополнительные параметры	
	DNAT (Destination Network Address Translation)	
–to-destination	Указывает, какой IP адрес должен быть подставлен в качестве адреса места назначения. В примере во всех пакетах протокола tcp, пришедших на адрес 1.2.3.4, данный адрес будет заменен на 4.3.2.1.	iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 -j DNAT –to-destination 4.3.2.1
	LOG	
–log-level	Используется для задания уровня журналирования (log level). В примере установлен максимальный уровень логирования для всех tcp пакетов в таблице filter цепочки FORWARD.	iptables -A FORWARD -p tcp -j LOG –log-level debug
–log-prefix	Задает текст (префикс), которым будут предваряться все сообщения iptables. Префикс может содержать до 29 символов, включая и пробелы. В примере отправляются в syslog все tcp пакеты в таблице filter цепочки INPUT с префиксом INRUT-filter.	iptables -A INPUT -p tcp -j LOG –log-prefix INRUT-filter
–log-ip-options	Позволяет заносить в системный журнал различные сведения из заголовка IP пакета.	iptables -A FORWARD -p tcp -j

Основные цепочки межсетевого экрана Netfilter:

• PREROUTING — изначальная обработка входящих пакетов

• INPUT — для входящих пакетов, адресованных непосредственно локальному компьютеру

• FORWARD — для маршрутизируемых пакетов

• OUTPUT — для пакетов, исходящих с локального компьютера

• POSTROUTING — для окончательной обработки исходящих пакетов

Таблицы межсетевого экрана Netfilter:

• raw - используется для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепочках PREROUTING и OUTPUT.

• mangle — содержит правила модификации IP-пакетов.

• nat - предназначена для подмены адреса отправителя или получателя. Данную таблицу проходят только первые пакеты из потока - трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT, и POSTROUTING.

• filter — основная таблица, используется по умолчанию если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD, и OUTPUT.

Пример создания правила для межсетевого экрана

Рассмотрим две цепочки, задающие два основных правила Iptables — PREROUTING и FORWARD.

• iptables -t nat -A PREROUTING -i eth0 -j DNAT —to-destination 192.168.57.102

• iptables -A FORWARD -d 192.168.57.102 -j ACCEPT

Первая из них определяет первоначальную обработку всех пакетов, приходящих на адаптер eth0:

• -t определяет подключаемую таблицу, в данном случае — nat — для подмены адреса отправителя или получателя

• -A — выбор цепочки

• -i — входящий интерфейс

• -j — действие с пакетами, удовлетворяющими условию — в данном случае DNAT — подмена адреса получателя

• –to-destination — выбор адреса, на оторый перенаправляются пакеты

• Вторая определяет проброс пакетов через сервер:

• -A — выбор цепочки

• -d — выбор адресата

• -j — выбор действия

Web Application Firewall

WAF (Web Application Firewall) - это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.

Основные преимущества:

• Анализ поведения пользователя в используемом приложении;

• Позволяет осуществлять мониторинг HTTP трафика и проводить анализ событий в реальном режиме времени;

• Предотвращение вредоносных запросов;

• Распознавание большинства опасных угроз;

• Дополнение сетевых средств безопасности;

• Просматривать детальные отчеты об атаках и попытках взлома.

•

Задания к лабораторной работе

Часть 1

• Установите web-сервер <sudo apt-get install apache2>

• Просмотрите список текущих правил iptables таблицы filter

sudo iptables -L

• Вы увидите, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и FORWARD), в каждой из которых установлена политика по умолчанию (на данный момент это ACCEPT).

• С помощью команды <sudo iptables -S> данный список можно просмотреть в другом формате, который отражает команды, необходимые для активации правил и политик.

• Чтобы сбросить текущие правила (если таковые есть), наберите:

sudo iptables -F

• Цепочка INPUT отвечает за входящий траффик.

• Чтобы внести локальный интерфейс выполните:

sudo iptables -A INPUT -i lo -j ACCEPT

• Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавьте два порта (порт SSH 22 и порт http 80), что разрешит трафик на эти порты.

sudo iptables -A INPUT -p tcp -m tcp –dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp -m tcp –dport 80 -j ACCEPT

• В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого:

sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT

• Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т.е. использовать ping или запускать обновления программного обеспечения):

sudo iptables -I INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

• Создав все эти правила, можно заблокировать все остальное и разрешить все исходящие соединения.

sudo iptables -P OUTPUT ACCEPT
sudo iptables -P INPUT DROP

• Просмотрите список правил

sudo iptables -L

• Добавим еще несколько правил для блокировки наиболее распространенных атак. Для начала нужно заблокировать нулевые пакеты <sudo iptables -A INPUT -p tcp –tcp-flags ALL NONE -j DROP>.

• Следующее правило отражает атаки syn-flood <sudo iptables -A INPUT -p tcp ! –syn -m state –state NEW -j DROP>. Теперь фаервол не будет принимать входящих пакетов с tcp-флагами. Нулевые пакеты, по сути, разведывательные. они используются, чтобы выяснить настройки сервера и определить его слабые места.

• Далее нужно защитить сервер от разведывательных пакетов XMAS <sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP>. Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.

• Со второй виртуальной машины, на которую установите nmap, проведите XMAS сканирование <sudo nmap -sX>.

• По умолчанию все несохраненные правила действуют до следующей перезагрузки сервера; сразу же после перезагрузки несохраненные правила будут потеряны. Самый простой способ загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. Во время инсталляции пакет уточнит, нужно ли сохранить текущие правила для дальнейшей автоматической загрузки, если текущие правила были протестированы и соответствуют всем требованиям, их можно сохранить.

Часть 2

• Для начала понадобится LAMP(Apache, MySQL, PHP). В лабораторной работе № 8, уже было показано, как установить его, используя tasksel.

• Установите mod_security <sudo apt-get install libapache2-mod-security2>

• Выполните команду <sudo apachectl -M | grep –color security2>. Если на экране появился модуль по имени security2_module (shared), значит, все прошло успешно.

• В каталоге логов Apache можно найти новый лог-файл для mod_security. /var/log/apache2/modsec_audit.log

• Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать: <sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity /modsecurity.conf>.

• Затем перезапустите Apache <sudo service apache2 reload>.

• Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируйте файл modsecurity.conf: <sudo nano /etc/modsecurity/modsecurity.conf>

• Найдите в файле строку: “SecRuleEngine DetectionOnly”. И измените ее так: “SecRuleEngine On”.

• Найдите “SecResponseBodyAccess On” и замените на “SecResponseBodyAccess Off”. Эта директива отвечает за буферизацию тела ответа; ее рекомендуется включать, только если требуется обнаружение и предохранение от утечки данных. Включенная директива (SecResponseBodyAccess On) не только будет использовать больше ресурсов сервера, но и увеличит размер лог-файла, следовательно, ее желательно отключить.

• По умолчанию mod_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/

• Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируйте файл mod-security.conf:

nano /etc/apache2/mods-enabled/mod-security.conf

• Между <IfModule security2_module> </IfModule> внесите следующие параметры:

```
Include "/usr/share/modsecurity-crs/*.conf"
Include "/usr/share/modsecurity-crs/activated_rules/*.conf"
```

• Директория activated_rules аналогична директории Apache mods-enabled. Правила доступны в каталогах: /usr/share/modsecurity-crs/base_rules ; /usr/share/modsecurity-crs/optional_rules ; /usr/share/modsecurity-crs/experimental_rules

• Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated_rules. <cd /usr/share/modsecurity-crs/activated_rules/>

• Добавьте несколько правил, например <sudo ln -s /usr/share/modsecurity-crs/base_rules/modsecurity_crs_30_http_policy.conf> ; <sudo ln -s /usr/share/modsecurity-crs/base_rules/modsecurity_crs_49_generic_attacks.conf>

• Чтобы новые правила вступили в исполнение, нужно перезапустить Apache <sudo service apache2 reload>

Вопросы к лабораторной работе

1. Что такое межсетевой экран?
2. Для чего используется межсетевой экран?
3. Принцип работы Netfilter.
4. Таблицы межсетевого экрана Netfilter. Для чего они используются?
5. Что такое правила межсетевого экрана?
6. Как создавать правила для межсетевого экрана утилитой Iptables?
7. Как сохранить правила для последующей автозагрузки?
8. Что такое Web Application Firewall?
9. Как настроить правила в WAF mod_security?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

3.Лабораторная работа № 3 Построение и управление RAID – массивами и логическими томами.

Цель работы: Получение теоретических и практических навыков построения и управления RAID массивами и логическими томами.

3.1. Теоретическая часть

Консольные команды:

- mdadm <параметры> - Консольная программа управления программными RAID массивами в Linux.
- lvm <параметры> - Консольная программа управления логическими томами LVM.
- parted <параметры> - Консольная программа для управления дисками
- watch <параметры> - Консольная программа, которая позволяет следить за изменениями в выводе команды.

RAID

RAID (Redundant Array of Independent Disks - избыточный массив независимых жестких дисков) - массив, состоящий из нескольких дисков, управляемых программным или аппаратным контроллером, связанных между собой и воспринимаемых как единое целое. В зависимости от того, какой тип массива используется, может обеспечивать различные степени быстродействия и отказоустойчивости. Служит для повышения надежности хранения данных и/или для повышения скорости чтения/записи информации.

Калифорнийский университет в Беркли предложил следующие уровни спецификации RAID, которые являются стандартом во всем мире:

- RAID 0 представлен как дисковый массив повышенной производительности, без отказоустойчивости. (Требуется минимум 2 диска)
- RAID 1 определен как зеркальный дисковый массив. (Требуется минимум 2 диска)
- RAID 2 массивы, в которых применяется код Хемминга. (Требуется минимум 7 дисков, для рационального использования)
- RAID 3 и 4 используют массив дисков с чередованием и выделенным диском четности. (Требуется минимум 4 диска)
- RAID 5 используют массив дисков с чередованием и “невыделенным диском четности”. (Требуется минимум 3 диска)
- RAID 6 используют массив дисков с чередованием и двумя независимыми “четностями” блоков. (Требуется минимум 4 диска)
- RAID 10 - RAID 0, построенный из RAID 1 массивов. (Требуется минимум 4 диска, четное количество)
- RAID 50 - RAID 0, построенный из RAID 5 массивов. (Требуется минимум 6 дисков, четное количество)
- RAID 60 - RAID 0, построенный из RAID 6 массивов. (Требуется минимум 8 дисков, четное количество)

Пример создания RAID 10

Проверим наличие виртуальных дисков.

```
sit@sit:~$ sudo parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  256MB   255MB   primary   ext2         boot
 2      257MB   21.5GB  21.2GB  extended
 5      257MB   21.5GB  21.2GB  logical               lvm

Error: /dev/sdb: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdc: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdd: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdd: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sde: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sde: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdf: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdf: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/sit--vg-swap_1: 533MB
```

Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number Start End   Size   File system   Flags
 1   0.00B 533MB 533MB linux-swap(v1)

Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/sit--vg-root: 20.7GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
Number Start End   Size   File system Flags
 1   0.00B 20.7GB 20.7GB ext4
sit@sit:~$

Примечание

Как видно из листинга, у нас присутствуют диски sda (на котором установлена операционная система Linux),sdb,sdc,sdd,sde,sdf. Теперь можно построить массив RAID 10 из дисков sdb, sdc, sdd и sde, а диск sdf пометим как диск горячей замены (применяется для горячей замены в случае отказа одного из дисков RAID массива).

Предупреждение

Необходимо открыть два терминала. В одном создается RAID массив, в другом осуществляется процесс наблюдения за созданием RAID массива.

Запустим процесс отслеживания состояния RAID массивов в терминале №1:

sit@sit:~$ sudo watch -n1 cat /proc/mdstat

Создадим RAID 10 в отдельном терминале №2:

sit@sit:~$ sudo mdadm -C /dev/md0 -l 10 -n 4 -x 1 /dev/sd[b-f]
[sudo] password for sit:
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
sit@sit:~$

В терминале №1 наблюдаем процесс создания RAID 10:

Every 1.0s: cat /proc/mdstat   Wed Sep 23 18:02:03 2015
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdf[4](S) sde[3] sdd[2] sdc[1] sdb[0]
16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
  [============>.....] resync = 61.3% (10286144/16760832) finish=0.5min speed=201781K/sec
unused devices: <none>

Создадим раздел в 1GB с файловой системой ext4 на созданном RAID 10:

sit@sit:~$ sudo parted /dev/md0
[sudo] password for sit:
GNU Parted 3.2
Using /dev/md0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel
New disk label type? GPT
Warning: The existing disk label on /dev/md0 will be destroyed and all data on this disk will be lost. Do you want to continue?
Yes/No? yes
(parted) mkpart

Partition name? []?
File system type? [ext2]? ext4
Start? 0
End? 1GB
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 17.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start   End   Size   File system Name Flags
 1   17.4kB 1000MB 1000MB ext4
(parted)

Отформатируем созданный раздел в файловую систему ext4:

sit@sit:~$ sudo mkfs.ext4 /dev/md0p1

Смонтируем созданный раздел:

sudo mount -t ext4 /dev/md0p1 /mnt/

Скопируем файлы на раздел с файловой системой ext4:

sudo cp -R /var/log/* /mnt/

Разрушим один диск и проверим целостность данных.:

Наблюдаем процесс как диск горячей замены встает на место сбойного диска

Every 1.0s: cat /proc/   Wed Sep 23 19:52:04 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdf[4] sde[3] sdd[2] sdc[1] sdb[0](F)
16760832 blocks super 1.2 512K chunks 2 near-copies [4/3] [_UUU]
[====>.............] recovery = 21.8% (1832192/8380416) finish=0.4min speed=229024K/sec
unused devices: <none>

Убедимся в целостности данных на разделе:

sit@sit:~$ ls -la /mnt/
total 968
drwxr-xr-x 9 root root   4096 Sep 23 19:34 .
drwxr-xr-x 22 root root   4096 Sep 19 14:26 ..
-rw-r--r-- 1 root root 18625 Sep 23 19:34 alternatives.log
drwxr-xr-x 2 root root   4096 Sep 23 19:34 apt
-rw-r----- 1 root root 41820 Sep 23 19:34 auth.log
-rw-r--r-- 1 root root 63653 Sep 23 19:34 bootstrap.log
-rw------- 1 root root   0 Sep 23 19:34 btmp
drwxr-xr-x 2 root root   4096 Sep 23 19:34 dist-upgrade
-rw-r----- 1 root root   31 Sep 23 19:34 dmesg
-rw-r--r-- 1 root root 339677 Sep 23 19:34 dpkg.log
-rw-r--r-- 1 root root 32032 Sep 23 19:34 faillog
drwxr-xr-x 2 root root   4096 Sep 23 19:34 fsck
drwxr-xr-x 3 root root   4096 Sep 23 19:34 installer
-rw-r----- 1 root root 189514 Sep 23 19:34 kern.log
drwxr-xr-x 2 root root   4096 Sep 23 19:34 landscape
-rw-r--r-- 1 root root 292292 Sep 23 19:34 lastlog
drwx------ 8 root root 16384 Sep 23 19:32 lost+found

-rw-r----- 1 root root 173386 Sep 23 19:34 syslog
-rw-r----- 1 root root   3090 Sep 23 19:34 syslog.1
-rw-r----- 1 root root   591 Sep 23 19:34 syslog.2.gz
-rw-r----- 1 root root 30788 Sep 23 19:34 syslog.3.gz
drwxr-x--- 2 root root   4096 Sep 23 19:34 unattended-upgrades
-rw-r--r-- 1 root root   8832 Sep 23 19:34 wtmp
sit@sit:~$ sudo head -n 10 /mnt/auth.log
Sep 19 14:38:02 sit systemd-logind [506]: Watching system buttons on /dev/input/event0 (Power Button)
Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event1 (Sleep Button)
Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event5 (Video Bus)
Sep 19 14:38:02 sit systemd-logind[506]: New seat seat0.
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event0 (Power Button)
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event1 (Sleep Button)
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event6 (Video Bus)
Sep 19 14:40:10 sit systemd-logind[508]: New seat seat0.
Sep 19 14:40:27 sit login[529]: pam_unix(login:session): session opened for user sit by LOGIN(uid=0)
Sep 19 14:40:27 sit systemd-logind[508]: New session c1 of user sit.

sit@sit:~$ sudo head -n 10 /mnt/syslog
Sep 23 07:17:01 sit CRON[2263]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 08:17:01 sit CRON[2266]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 09:17:01 sit CRON[2269]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 10:17:01 sit CRON[2272]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 10:46:05 sit dhclient: DHCPREQUEST of 10.0.2.15 on eth0 to 10.0.2.2 port 67 (xid=0x6a9a8b24)
Sep 23 10:46:05 sit dhclient: DHCPACK of 10.0.2.15 from 10.0.2.2
Sep 23 10:46:05 sit dhclient: bound to 10.0.2.15 -- renewal in 42505 seconds.
Sep 23 11:17:01 sit CRON[2285]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 12:17:01 sit CRON[2288]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 13:17:01 sit CRON[2291]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)

Сделаем имитацию замены извлечением и вставки нового диска.:

sit@sit:~$ sudo mdadm /dev/md0 -r /dev/sdb
mdadm: hot removed /dev/sdb from /dev/md0
sit@sit:~$ sudo mdadm /dev/md0 -a /dev/sdb
mdadm: added /dev/sdb
sit@sit:~$

Наблюдаем что диск sdb пометился как диск горячей замены.:

Every 1.0s: cat /proc/ Wed Sep 23 19:59:09 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb[5](S) sdf[4] sde[3] sdd[2] sdc[1]
16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>

Примечание

Для того чтобы остановить RAID используется параметр **–stop** команды mdadm.

Для очистки записи принадлежности к программному RAID используется параметр **–zero-superblock** команды mdadm.

LVM

LVM (Logical Volume Manager) - менеджер логических томов является уникальной системой управления дисковым пространством. Она позволяет с легкостью использовать и эффективно управлять дисковым пространством. Уменьшает общую нагруженность и сложность существующей системы. У логических томов, которые созданы через LVM, можно легко изменять размер, а названия, которые им даны, помогут в дальнейшем определить назначение тома.

- PV, Physical Volume или физический том. Чаще всего это раздел на диске или весь диск. К ним относят устройства программного и аппаратного RAID массивов (которые могут включать в себя еще несколько физических дисков). Физические тома объединяются и образуют группы томов.

- VG, Volume Group или группа томов. Это самый верхний уровень модели представления, которая используется в LVM. С одной стороны группа томов может состоять из физических томов, с другой- из логических томов и представлять собой единую структуру.

- LV, Logical Volume или логический том. Раздел в группе томов, тоже самое, что раздел диска в не-LVM системе. Является блочным устройством и, как следствие, может содержать файловую систему.

- PE, Physical Extent или физический экстент. Каждый физический том делится на блоки данных - физические экстенты. Они имеют размеры как и у логических экстентов.

- LE, Logical Extent или логический экстент. Каждый логический том также делится на блоки данных - логические экстенты. Размеры логических экстентов не меняются в рамках группы томов.

Инициализация дисков и разделов

Перед тем, как начать использовать диск или раздел в качестве физического тома, важно его проинициализировать. Осуществляется это с помощью команды **pvcreate**. Данная команда создаст в начале диска или раздела дескриптор группы томов.

Для диска:

sit@sit:~$ sudo pvcreate /dev/sdb
[sudo] password for sit:
Physical volume "/dev/sdb" successfully created

Для разделов:

sit@sit:~$ sudo pvcreate /dev/sdb1
[sudo] password for sit:
Physical volume "/dev/sdb1" successfully created

Примечание

Повторяем данную операцию для всех дисков или разделов которые необходимо поменять как физические тома LVM.

В нашем случае это - sdb, sdc , sde, sdd, sdf.

Предупреждение

Если появилась ошибка инициализации диска с таблицей разделов, проверьте, что работаете с нужным диском. Убедившись в этом выполните следующие команды:

sudo dd if=/dev/zero of=/dev/sd* bs=1k count=1

sudo blockdev -rereadpt /dev/sd*

*Данные команды уничтожат существующую таблицу разделов на диске sd**. Для разделов воспользуйтесь утилитой fdisk (parted или gdisk) и установите тип раздела в 0x8e (LVM).

Просмотреть диски (разделы) которые помечены как физические тома LVM можно с помощью команды **pvdisplay**.

```
sit@sit:~$ sudo pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               dt4vrH-xpIo-IOAR-4sZD-Q9cT-St7Q-dRKInS

  --- Physical volume ---
  PV Name               /dev/sdc
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               TD4x9x-t6dp-vrJ9-GnKk-eX1J-bU06-L17fnt

  --- Physical volume ---
  PV Name               /dev/sdd
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               qgJYg6-fNAu-9P2v-lBvt-u1H5-lfml-Pb186U

  --- Physical volume ---
  PV Name               /dev/sde
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
```

```
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               bKGRsE-ZNNV-XtqW-bXpn-yOI1-DMdC-8rANuv

  --- Physical volume ---
  PV Name               /dev/sdf
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               W6TBLw-3Yt6-ZJE2-lcOb-PMni-F95G-lxmyHW
```

Создание группы томов.

Для создания группы томов необходимо воспользоваться командой **vgcreate**. На вход программы необходимо указать имя группы и диски (разделы) которые необходимо добавить в данную группу.

```
sit@sit:~$ sudo vgcreate storage /dev/sd[b-f]
  Volume group "storage" successfully created
```

Просмотреть группы томов в системе можно с помощью команды **vgdisplay**.

```
sit@sit:~$ sudo vgdisplay
  --- Volume group ---
  VG Name               storage
  System ID
  Format                lvm2
  Metadata Areas        5
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                5
  Act PV                5
  VG Size               39.98 GiB
  PE Size               4.00 MiB
  Total PE              10235
  Alloc PE / Size       0 / 0
  Free  PE / Size       10235 / 39.98 GiB
  VG UUID               Nf04a2-sQ5O-zRfO-V3jc-wpTj-KjYx-aKpeCK
```

Удаление группы томов.

Для удаления группы томов необходимо убедиться, что целевая группа томов не содержит логических томов. Далее необходимо деактивировать группу томов

```
sudo vgchange -an storage
```

После чего удалить группу томов командой

```
sudo vgremove storage
```

Примечание

Для того, чтобы добавить ранее инициализированный физический том в существующую группу томов используется команда **vgextend**

sudo vgextend storage /dev/sd*

Для того, чтобы удалить физический том из группы томов необходимо воспользоваться командой **vgreduce**

sudo vgreduce storage /dev/sd*

Создание логического тома.

Для того, чтобы например создать логический том “sit”, размером 1800Мб, необходимо выполнить команду

sudo lvcreate -L1800 -n sit storage

Примечание

Без указания суффикса размеру раздела, по умолчанию используется множитель M «мегабайт» (в системе СИ равный 10^6 байт), что показано в примере выше. Суффиксы в верхнем регистре - KMGTPE соответствуют единицам в системе СИ с основанием 10. Например, G — гигабайт равен 10^9 байт, а суффиксы в нижнем регистре - kmgtpe соответствуют единицам в системе IEC (с основанием 2), например g — гибибайт равен 2^{30}байт.

Для того, чтобы создать логический том размером 100 логических экстентов с записью по двум физическим томам и размером блока данных в 4 KB

sudo lvcreate -i2 -I4 -l100 -n sit storage

Если необходимо создать логический том, который будет полностью занимать группу томов, то сперва используйте команду vgdisplay, чтобы узнать полный размер группы томов, а после этого выполните команду **lvcreate**.

sudo vgdisplay storage | grep "Total PE"

Total PE 10230

sudo lvcreate -l 10230 storage -n sit

Эти команды создают логический том sit, полностью заполняющий группу томов. Тоже самое можно реализовать командой

lvcreate -l100%FREE storage -n sit

Удаление логических томов.

Перед удалением логический том должен быть размонтирован

sudo umount /dev/storage/sit

sudo lvremove /dev/storage/sit

lvremove -- do you really want to remove "/dev/storage/sit"? [y/n]: y

lvremove -- doing automatic backup of volume group "storage"

lvremove -- logical volume "/dev/storage/sit" successfully removed

Увеличение логических томов.

Для того, чтобы увеличить логический том, необходимо указать команде lvextend размер, до которого будет увеличен том (в экстентах или в размере)

sudo lvextend -L15G /dev/storage/sit

lvextend -- extending logical volume "/dev/storage/sit" to 15 GB

lvextend -- doing automatic backup of volume group "storage"

lvextend -- logical volume "/dev/storage/sit" successfully extended

В результате /dev/storage/sit увеличится до 15Гбайт.

Примечание

Для изменения размера файловых систем ext2, ext3 и ext4 используйте **resize2fs**.

Создание снап-шотов LVM

Для того, чтобы создать снапшот необходимо использовать **lvcreate -s**

sudo lvcreate -s -L10GB -n backup /dev/storage/sit

Таким образом мы создадим снапшот в 10 GB с именем backup для хранения изменений.

Задания к лабораторной работе
Часть 1

- Добавить пять виртуальных жестких дисков.
- Запустить Linux.
- Установить mdadm.
- Ознакомится с утилитой mdadm, ее возможностями и параметрами.
- В отдельном терминале следить за состоянием файла /proc/mdstat
- Собрать RAID 1 с помощью mdadm.
- Создать на созданном RAID файловую систему ext4.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 1.
- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 0 с помощью mdadm.
- Создать на созданном RAID файловую систему ext3.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 0.
- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 5 с диском горячей замены с помощью mdadm.
- Создать на созданном RAID файловую систему ext4.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить три диска RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 5.

- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 10 с диском горячей замены с помощью mdadm.
- Создать на созданном RAID файловую систему ext2.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить два диска RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 10.
- Очистить информацию дисков о принадлежности к программному RAID.

Часть 2

- Инициализировать физические диски, поверх которых будет создан LVM.
- Создать группу томов на основе четырех виртуальных жестких дисков.
- Создать логический том.
- На созданном логическом томе создать файловую систему.
- Смонтировать систему и создать файл LVM.txt .
- Добавить в группу томов еще один виртуальный жесткий диск.
- Определить количество добавленных экстентов.
- Расширить созданный логический том на размер добавленных экстентов.
- Увеличить размер файловой системы.
- Сделать снапшот логического тома.
- Удалить группу томов и снапшот.

Вопросы к лабораторной работе

1. В чем достоинства и недостатки основных уровней RAID?
2. Какие основные команды использовались в лабораторной работе и каковы их назначения?
3. В чем смысл каждого из основных уровней RAID?
4. Сколько минимально необходимо дисков для каждого из основных уровней RAID?
5. Сколько максимально может выйти из строя дисков в основных уровнях RAID-массивов без потери данных?
6. Порядок действий для создания логического тома.

7. Что такое Snapshot? Как его создать, и какое его функциональное назначение в LVM.

8. Что такое экстенты в LVM, какое их функциональное назначение.

9. В чем отличие логического тома от физического?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

4.Лабораторная работа №4. Программное восстановление данных.

Цель работы: Получение теоретических и практических навыков программного восстановления данных..

4.1. Теоретическая часть

Восстановление данных TestDisk

TestDisk — свободная программа для восстановления данных, предназначенная прежде всего для восстановления потерянных разделов на носителях информации, а также для восстановления загрузочного сектора, после программных или человеческих ошибок (например, потеря MBR).

- Установка **<sudo apt-get install testdisk>**.
- Запускаем TestDisk **<sudo testdisk>**.
- Появляется окошко приветствия TestDisk, нам предлагается вести лог работы (для выполнения данной работы лог не требуется).
- Выбираем нужный диск и нажимаем **Enter**.
- Предлагается выбрать тип таблицы разделов, обычно TestDisk определяет все правильно, так что нажимаем **Enter**.
- Выбираем **Analise**.
- Выбираем **QuickSearch**.
- Нам выводят таблицу разделов. Выбираем раздел и нажимаем **P**, чтобы вывести список файлов.
- Выбираем файлы для восстановления и нажимаем **C**.
- Выбираем папку, куда будут сохранены файлы и нажимаем **C**.

Восстановление данных PhotoRec

PhotoRec - это утилита, входящая в состав пакета TestDisk. Предназначена для восстановления испорченных файлов с карт памяти цифровых фотоаппаратов (CompactFlash, Secure Digital, SmartMedia, Memory Stick, Microdrive, MMC), USB flash-дисков, жестких дисков и CD/DVD. Восстанавливает файлы большинства распространенных графических форматов, включая JPEG, аудио-файлы, включая MP3, файлы документов в форматах Microsoft Office, PDF и HTML, а также архивы, включая ZIP. Может работать с файловыми системами ext2, ext3, ext4 FAT, NTFS и HFS+, причем способна восстановить графические файлы даже в том случае, когда файловая система повреждена или отформатирована.

- Установка **<sudo apt-get install testdisk>**.
- Запускаем PhotoRec **<sudo photorec>**.
- Выбираем нужный диск и нажимаем **Enter**.
- В нижнем меню можно выбрать **File Opt**, чтобы выбрать типы файлов для восстановления (по умолчанию выбраны все).
- Чтобы начать восстановление нажмите **Enter**, выбрав **Search**.

- У нас выбрана система ext4, поэтому выбираем первый вариант [ext2/ext3].

- Если выбрать пункт **FREE**, то поиск будет произведен в пустом пространстве и в этом случае будут восстановлены только удаленные файлы, а если выбрать **WHOLE**, то поиск будет произведен на всем диске.

- Теперь нужно указать директорию, куда будем сохранять нужные нам файлы. Выбираем нужную папку и нажимаем **C**.

- Выбираем файлы для восстановления и нажимаем **C**.

Восстановление данных Extundelete

Extundelete – утилита, позволяющая восстанавливать файлы, которые были удалены с разделов ext3/ext4.

- Установка: **<sudo apt-get install extundelete>**.

- Как только вы поняли, что удалили нужные файлы, необходимо отмонтировать раздел:**<umount /dev/<partition> >**

- Зайдите в каталог, в который будут восстанавливаться удаленные данные. Он должен быть расположен на разделе отличном от того, на котором хранились восстанавливаемые данные: **cd /<путь_к_каталогу_куда_восстанавливать_данные>**

- Запустите **extundelete**, указав раздел, с которого будет происходить восстановление и файл, который необходимо восстановить: **sudo extundelete /dev/<partition> –restore-file /<путь к файлу>/<имя_файла>**

- Можно так же восстанавливать содержимое каталогов: **sudo extundelete /dev/<partition> –restore-directory /<путь_к_директории>**

Восстановление данных Foremost.

Foremost - консольная программа, позволяющая искать файлы на дисках или их образах по hex-данным, характерным заголовкам и окончаниям. Программа проверяет файлы на предмет совпадения заранее определённых hex-кодов (сигнатур), соответствующих наиболее распространённым форматам файлов. После чего экстрагирует их из диска/образа и складывает в каталог, вместе с подробным отчётом о том, чего, сколько и откуда было восстановлено. Типы файлов, которые foremost может сразу восстановить: jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, htm, cpp. Есть возможность добавлять свои форматы (в конфигурационном файле /etc/foremost.conf), о которых программа не знает.

- Установка: **<sudo apt-get install foremost>**

- Пример использования для восстановления изображений с диска **/dev/sdb** в каталог**~/out_dir**: **<sudo foremost -t jpg,gif,png,bmp -i /dev/sdb -o ~/out_dir>**

-

Задания к лабораторной работе

- Добавьте в виртуальную машину виртуальный жесткий диск.

- Запустите виртуальную машину с Linux.

- Запустите fdisk (gdisk или parted) и создайте таблицу разделов MBR с разделами.

- Отформатируйте созданные разделы в файловую систему ext4.
- Установите TestDisk.
- Удалите MBR (или таблицу разделов) с помощью команды DD.
- Восстановите MBR (или таблицу разделов) с помощью TestDisk.
- Смонтируйте восстановленные разделы и создайте там произвольные файлы.
- Удалите созданные файлы.
- С помощью TestDisk восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /var/log/ .
- Удалите данные с созданного каталога.
- С помощью PhotoRec восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /etc/ .
- С помощью Extundelete или Foremost восстановите данные.

Вопросы к лабораторной работе

1. С помощью какой из программ, используемых в этой лабораторной работе, можно восстановить таблицу разделов?
2. Какие файловые системы поддерживает PhotoRec?
3. Какие форматы поддерживает PhotoRec?
4. Как Foremost восстанавливает файлы?
5. Можно ли восстановить данные с файловой системы NTFS, используя extundelete?
6. Все ли данные скопированные с каталога /var/log/ восстановились?
7. Все ли данные скопированные с каталога /etc/ восстановились?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

2.5.Лабораторная работа № 5. Обнаружение и предотвращение вторжений.

Цель работы: - Получить сведения о том, как осуществляется защита с помощью систем обнаружения и предотвращения вторжений. Научиться использовать SNORT.

5.1. Теоретическая часть

Система обнаружения вторжений (IDS) — программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет.

Сетевая система обнаружения вторжений (англ. network intrusion detection system, NIDS) — система обнаружения вторжений, которая отслеживает такие виды вредоносной деятельности, как DoS атаки, сканирование портов или даже попытки проникновения в сеть.

В пассивной IDS при обнаружении нарушения безопасности, информация о нарушении записывается в лог приложения, а также сигналы опасности отправляются на консоль и/или администратору системы по определенному каналу связи. В активной системе, также известной как Система Предотвращения Вторжений (IPS — Intrusion Prevention system (англ.)), IDS ведет ответные действия на нарушение, сбрасывая соединение или перенастраивая межсетевой экран для блокирования трафика от злоумышленника. Ответные действия могут проводиться автоматически либо по команде оператора.

Обнаружение проникновения позволяет организациям защищать свои системы от угроз, которые связаны с возрастанием сетевой активности и важностью информационных систем. При понимании уровня и природы современных угроз сетевой безопасности, вопрос не в том, следует ли использовать системы обнаружения проникновений, а в том, какие возможности и особенности систем обнаружения проникновений следует использовать.

Snort — свободная сетевая система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом, способная выполнять регистрацию пакетов и в реальном времени осуществлять анализ трафика в IP-сетях.

Выполняет протоколирование, анализ, поиск по содержимому, а также широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как попытки атак на переполнение буфера, скрытое сканирование портов, атаки на веб-приложения, SMB-зондирование и попытки определения операционной системы. Программное обеспечение в основном используется для предотвращения проникновения, блокирования атак, если они имеют место.

Snort использует правила, написанные простым , но в то же время гибким и достаточно мощным языком. Существует ряд общих принципов написания, запомнить которые достаточно просто.

Большая часть правил Snort умещается в 1 строку. Это следствие того, что до версии 1.8 нельзя было использовать многострочные записи. В более поздних версиях правила можно растягивать на несколько строк, вставляя в конец строки символ “\” (без кавычек).

Правила Snort состоят из двух частей: заголовка правила и параметров правила. Заголовок содержит описание действия, протокол передачи данных, IP-адреса, сетевые маски и порты источника и назначения. Параметры правила хранят предупреждающее

сообщение, а также информацию о том, какую часть обнаруженного пакета нужно обработать в случае срабатывания правила.

Задания к лабораторной работе

• Узнайте свой ip адрес командой ifconfig

• Установите SNORT <sudo apt-get install snort>

• При установке будет нужно указать защищаемую сеть. Введите ..*.0/24 (Где ..* - первые три числа вашего ip-адреса, например этот будет 192.168.1.0/24, если вы используете VirtualBox и у вас в настройках сети стоит сетевой мост)

• Запустите SNORT <sudo service snort start>

• Настройка правил

• Перейдите в каталог /etc/snort/rules < cd /etc/snort/rules)

• Создайте файл с правилами <nano test.rules>

alert tcp any any -> any any (content:"https://www.google.ru/" ; msg:"Someone open Google website" ; sid: 12312313;)

• Перейдите в каталог /etc/snort <cd /etc/snort)

• Теперь нужно изменить содержимое конфигурационного файла Snort < sudo nano snort.conf>

• Найдите строчки с правилами (они начинаются с include $RULE_PATH, это в части Step 7) и добавьте файл с нашими правилами

include $RULE_PATH/test.tules

• В файле snort.conf так же укажите домашнюю сеть. В Step 1 измените строчку "ipvar HOME_NET any" , на

ipvar HOME_NET 192.168.1.0/24

• Запустите snort <sudo snort -A console -i eth0 -c snort.conf>

• Зайдите на https://www.google.ru/ и проверьте в терминале, как работает правило.

• Теперь нам понадобиться еще одна виртуальная машина, на ней должен быть установлен nmap.

• Со второй ВМ используйте ping, посмотрите, как реагирует SNORT

• Используйте различные методы сканирования nmap(используйте -sS, -sT, -sN, -sU, -sX, -sF и посмотрите, как реагирует SNORT;

• В файл test.rules добавьте правило обнаружения сканирования nmap -sN (NULL Scan)

alert tcp any any -> any any (msg:"NULL Scan"; flags: 0; sid:322222;)

• Запустите snort <sudo snort -A console -i eth0 -c snort.conf>

• Со второй виртуальной машины произведите NULL сканирование <sudo nmap -sN>, проверьте, как работает правило.

• Можно загрузить обновленные правила SNORT, для этого:

• Зарегистрируйтесь на сайте https://www.snort.org/ и скачайте последнюю версию правил

• Разархивируйте скачанный архив и скопируйте каталоги rules, so_rules и preproc_rules в /etc/snort :

```
sudo cp -R ./rules/ /etc/snort/
sudo cp -R ./so_rules/ /etc/snort/
sudo cp -R ./preproc_rules/ /etc/snort/
```

Вопросы к лабораторной работе

1. Что такое IDS?
2. Что такое сетевая система обнаружения вторжений?
3. Чем отличаются пассивные и активные IDS?
4. Что такое SNORT?
5. Какие задачи выполняет SNORT?
6. Как работают правила SNORT?
7. Как писать правила для SNORT?
8. Зачем писать собственные правила SNORT?
9. Зачем загружать обновление правил SNORT?
10. Как в SNORT создавать логи?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по практической работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: отчет.

6.Лабораторная работа № 6. Электронная цифровая подпись.

Цель работы:

Ознакомиться со схемами цифровой подписи и получить навыки создания и проверки подлинности ЦП.

6.1.Теоретическая часть

На протяжении многих веков при ведении деловой переписки, заключении контрактов и оформлении любых других важных бумаг подпись ответственного лица или исполнителя была непременным условием признания его статуса или неоспоримым свидетельством его важности. Подобный акт преследовал две цели:

- гарантирование истинности письма путем сличения подписи с имеющимся образцом;
- гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

- подпись аутентична, то есть с ее помощью получателю документа можно доказать, что она принадлежит подписывающему;
- подпись служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ, и никто другой не смог бы этого сделать;
- подпись непереносима, то есть является частью документа и поэтому перенести ее на другой документ невозможно:
- документ с подписью является неизменяемым, то есть после подписания его невозможно изменить, оставив данный факт незамеченным;
- подпись неоспорима, то есть человек, подписавший документ, в случае признания экспертизой, что именно он засвидетельствовал данный документ, не может оспорить факт подписания;
- любое лицо, имеющее образец подписи, может удостовериться в том, что данный документ подписан владельцем подписи.

С переходом к безбумажным способам передачи и хранения данных, а также с развитием систем электронного перевода денежных средств, в основе которых – электронный аналог бумажного платежного поручения, проблема виртуального подтверждения аутентичности документа приобрела особую остроту. Развитие любых подобных систем теперь немыслимо без существования электронных подписей под электронными документами. Однако применение и широкое распространение *электронно-цифровых подписей* (ЭЦП) повлекло целый ряд правовых проблем. Так, ЭЦП может применяться на основе договоренностей внутри какой-либо группы пользователей системы передачи данных, и в соответствии с договоренностью внутри данной группы ЭЦП должно иметь юридическую силу. Но будет ли электронная подпись иметь доказательную силу в суде, например, при оспаривании факта передачи платежного поручения?

Схема 1

Данная схема предполагает шифрование электронного документа (ЭД) на основе симметричных алгоритмов и предусматривает наличие в системе третьего лица (арбитра),

пользующегося доверием участников обмена подписанными подобным образом электронными документами. Взаимодействие пользователей данной системой производится по следующей схеме:

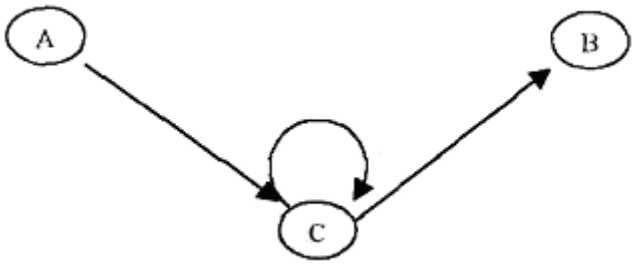


Рис. 5.1. Основные методы построения схем ЭЦП. Схема 1.

Участник А зашифровывает сообщение своим секретном ключе K_A, знание которого разделено с арбитром (С на рис. 1), затем шифрованное сообщение передается арбитру с указанием адресата данного сообщения (информация, идентифицирующая адресата, передается также в зашифрованном виде).

Арбитр расшифровывает полученное сообщение ключом K_A, производит необходимые проверки и затем зашифровывает его секретным ключом участника В (K_B). Далее зашифрованное сообщение посылается участнику В вместе с информацией, что оно пришло от участника А.

Участник В расшифровывает данное сообщение и убеждается в том, что отправителем является участник А.

Авторизацией документа в данной схеме считается сам факт шифрования электронного документа (ЭД) секретным ключом и передача зашифрованного ЭД арбитру. Основным преимуществом этой схемы является наличие третьей стороны, исключающей какие-либо спорные вопросы между участниками информационного обмена, то есть в данном случае не требуется дополнительной системы арбитража ЭЦП. Недостатком схемы является так же наличие третьей стороны и использование симметричных алгоритмов шифрования. На практике эта схема не получила широкого распространения.

Схема 2

Фактом подписания документа в данной схеме является шифрование документа секретным ключом его отправителя. Здесь используются асимметричные алгоритмы шифрования.

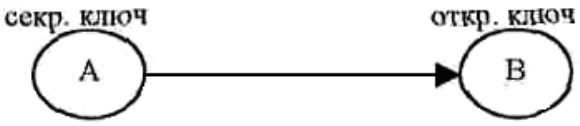


Рис. 5.2. Основные методы построения схем ЭЦП. Схема 2.

Вторая схема используется довольно редко вследствие того, что длина ЭД может оказаться очень большой (шифрование асимметричным алгоритмом может оказаться неэффективным по времени), но в этом случае в принципе не требуется наличие третьей стороны, хотя она и может выступать в роли сертификационного органа открытых ключей пользователя.

Схема 3

Наиболее распространенная схема ЭЦП использует шифрование окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма. Структурная схема такого варианта построения ЭЦП представлена на рисунке 3.

Рис.5.3. Основные методы построения схем ЭЦП. Схема 3.

Процесс генерации ЭЦП происходит следующим образом. Участник А вычисляет хэш-код от ЭД. Полученный хэш-код проходит процедуру преобразования с использованием своего секретного ключа. После чего полученное значение (которое и является ЭЦП) вместе с ЭД отправляется участнику В.

Участник В должен получить ЭД с ЭЦП и сертифицированный открытый ключ участника А, а затем произвести расшифрование на нем ЭЦП, сам ЭД подвергается операции хэширования, после чего результаты сравниваются, и если они совпадают, то ЭЦП признается истинной, в противном случае ложной.

В настоящее время применяются несколько алгоритмов цифровой подписи:

- RSA (наиболее популярен);
- Digital Signature Algorithm, DSA (алгоритм цифровой подписи американского правительства, который применяют в стандарте цифровой подписи (Digital Signature Standard, DSS), также используется часто);
- алгоритм Эль-Гамаля (иногда можно встретить).
- алгоритм, который применяют в стандарте ГОСТ Р34.10-94 (в основе лежит DSA и является вариацией подписи Эль-Гамаля);
- Так же существуют алгоритмы подписей, в основе которых лежит криптография эллиптических кривых; они похожи на все прочие, но в некоторых ситуациях работают эффективнее.

Электронная подпись RSA

Для осуществления подписи сообщения $m=m_1m_2m_3..m_n$ необходимо вычислить хеш-функцию $y=h(m_1m_2m_3..m_n)$, которая ставит в соответствие сообщению m число y. На следующем шаге достаточно снабдить подписью только число y, и эта подпись будет относиться ко всему сообщению m.

Далее по алгоритму RSA вычисляются ключи (e,n) и (d,n) (см. лабораторную работу №11 (4)).

Затем вычисляется (d на этот раз секретная степень).

Число s это и есть цифровая подпись. Она просто добавляется к сообщению и получается подписанное сообщение <m,s>.

Теперь каждый, кто знает параметры подписавшего сообщение (т.е. числа e и n), может проверить подлинность подписи.

Для этого необходимо проверить выполнение равенства .

Алгоритм Эль-Гамаля

Для генерации пары ключей сначала выбирается простое число p и два случайных числа g и x. Оба эти числа должны быть меньше p.

Чтобы подписать сообщение M, сначала выбирается случайное число k, взаимно простое с p-1. Затем вычисляется

$$a = g^k \bmod p$$

и с помощью расширенного алгоритма Евклида находится b в следующем уравнении:

$$M = (xa + kb) \bmod (p - 1)$$

Подписью является пара чисел: a и b. Случайное значение k должно храниться в секрете. Для проверки подписи нужно убедиться, что

$$y^a a^b \bmod p = g^M \bmod p$$

Открытый ключ:

p	простое число (может быть общим для группы пользователей)
g	$<p$ (может быть общим для группы пользователей)
y	$=g^x \bmod p$

Закрытый ключ:

x	$<p$

Подпись:

k	выбирается случайным образом, взаимно простое с p-1
a	(подпись) $=g^k \bmod p$
b	(подпись), такое что $M = (xa + kb) \bmod (p - 1)$

Проверка:

Подпись считается правильной, если $y^a a^b \bmod p = g^M \bmod p$

ПРИМЕР (алгоритм Эль-Гамаля)

1)Пусть общие параметры для некоторого сообщества пользователей p=23 и g=5.

Пусть секретный ключ x=7. Вычислим открытый ключ y:

2)

Пусть нужно поставить подпись на сообщение m=baaqab

Перейдем к вычислению подписи по алгоритму.

3)Прежде всего, вычисляется хеш-функция. Пусть её значение h(m)=h(baaqab)=M=3.

4)Затем генерируется случайное число k, например k=5. Вычисляем по формулам
5)
И по расширенному алгоритму Евклида находим b
6)3=(7*20+5*b) mod 22
Такое b существует, т.к. НОД(k,p-1)=1. Получили b=21.
7)Получили подписанное сообщение в виде <baaqab,20,21>
Подписанное сообщение передается.

Полученное сообщение проверим на подлинность.

1)Прежде всего, вычисляется хеш-функция h(baaqab)=M=3.
2)Затем вычисляем левую часть

$$y^a a^b \bmod p = g^M \bmod p$$

3)и после этого правую
Так как левая часть совпала с правой, то можно сделать вывод, что подпись верна.

Задание

Реализовать приложение, позволяющие решить задачи в соответствии с вариантом.

Задачи

1.Для указанных открытых ключей пользователя RSA проверить подлинность подписанных сообщений:

1) n=55, e=3: <7,28>, <22,15>,<16,36>

2) n=65, e=5: <6,42>, <10,30>,<6,41>

3) n=77, e=7: <13,41>, <11,28>,<5,26>

4) n=91, e=5: <15,71>, <11,46>,<16,74>

5) n=33, e=3: <10,14>, <24,18>,<17,8>

2. Абоненты некоторой сети применяют подпись Эль-Гамаля с общими параметрами p=23, g=5. Для указанных секретных параметров абонентов найти открытый ключ (y) и построить подпись для сообщения m:

1) x=11, k=3, m=15

2) x=10, k=15, m=5

3) x=3, k=13, m=8

4) x=18, k=7, m=5

5) x=9, k=19, m=15

Во всех вариантах будем предполагать, что h(m)=m для всех значений m.

№ вариант а	№№ задач
1	1.1 , 2.1
2	1.2 , 2.2
3	1.3 , 2.3
4	1.4 , 2.4
5	1.5 , 2.5
6	1.4 , 2.2
7	1.3 , 2.1

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по практической работе».

Контрольные вопросы

1. Основные методы построения схем ЭЦП.
2. Шифрование окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма.
3. Процессы генерации ЭЦП.
4. Электронная подпись RSA.
5. Алгоритм Эль-Гамаля..

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2 Акустоэлектромагнитный активный канал утечки речевой информации	1-2	1-3	1-3

Оценочные средства: отчет.

7.Лабораторная работа № 7. Программно-аппаратное шифрование данных при их хранении.

Цель работы:

Освоить шифрование данных с использованием шифрующей системы (Encrypting File System - EFS).

7.1.Теоретическая часть

Шифрующая *файловая система* (Encrypting File *System* - *EFS*) появилась в операционных системах семейства *Windows*, начиная с *Windows 2000*. Она позволяет шифровать отдельные папки и файлы на томах с файловой системой *NTFS*. Рассмотрим этот механизм подробнее.

Сначала несколько слов о рисках, которые можно снизить, внедрив данный механизм. Повышение мобильности пользователей приводит к тому, что большое количество конфиденциальных данных (предприятий или личных) оказывается на дисках ноутбуков, на съемных носителях и т.д. *Вероятность* того, что подобное устройство будет украдено или временно попадет в чужие руки, существенно выше чем, например, для жесткого диска корпоративного персонального компьютера (хотя и в этом случае, возможны кражи или *копирование* содержимого накопителей). Если данные хранить в зашифрованном виде, то даже если носитель украден, *конфиденциальность* данных нарушена не будет. В этом и заключается цель использования *EFS*.

Следует учитывать, что для передачи *по* сети, зашифрованный *EFS файл* будет расшифрован, и для защиты данных в этих случаях надо использовать дополнительные *механизмы*.

Рассмотрим работу *EFS*. Пусть, у нас имеется *сервер Windows Server* 2008, входящий в *домен*, и три учетные записи, обладающие административными правами на сервере (одна из них - встроенная административная *запись Administrator*).

Пользователь **User1** хочет защитить конфиденциальные файлы. Тут надо отметить, что хотя шифровать с помощью *EFS* можно и отдельные файлы, рекомендуется применять *шифрование* целиком к папке.

User1 с помощью оснастки **Certificates** запрашивает сертификат (можно выбрать *шаблон User* или *Basic EFS*). Теперь у него появляется ключевая пара и *сертификат открытого ключа*, и можно приступать к шифрованию.

Чтобы зашифровать папку, в ее свойствах на вкладке **General** нажимаем кнопку **Advanced** и получаем *доступ* к атрибуту, указывающему на*шифрование* файла.

Рис. 7.1. В свойствах папки устанавливаем шифрование

Работа *EFS* организована так, что одновременно сжатие и *шифрование* файлов и папок осуществляться не может. Поэтому нельзя разом установить атрибуты **Compress contents to save disk** и **Encrypt contents to secure data** (рис. 7.1).

При настройках *по* умолчанию, зашифрованная *папка* выделяется в проводнике зеленым цветом. Для зашифровавшего *файл* пользователя порядок работы с ним не изменится.

Теперь выполним "переключение пользователей" и зайдем в систему под другой учетной записью, обладающей административными правами, но не являющейся встроенной административной записью. Пусть это будет **User2**.

Несмотря на то, что **User2** имеет такие же разрешения на *доступ* к файлу, что и **User1**, прочитать он его не сможет (рис. 7.2).

Также он не сможет его скопировать, т.к. для этого надо расшифровать *файл*. Но надо учитывать, что **User2** может удалить или переименовать *файл* или папку.

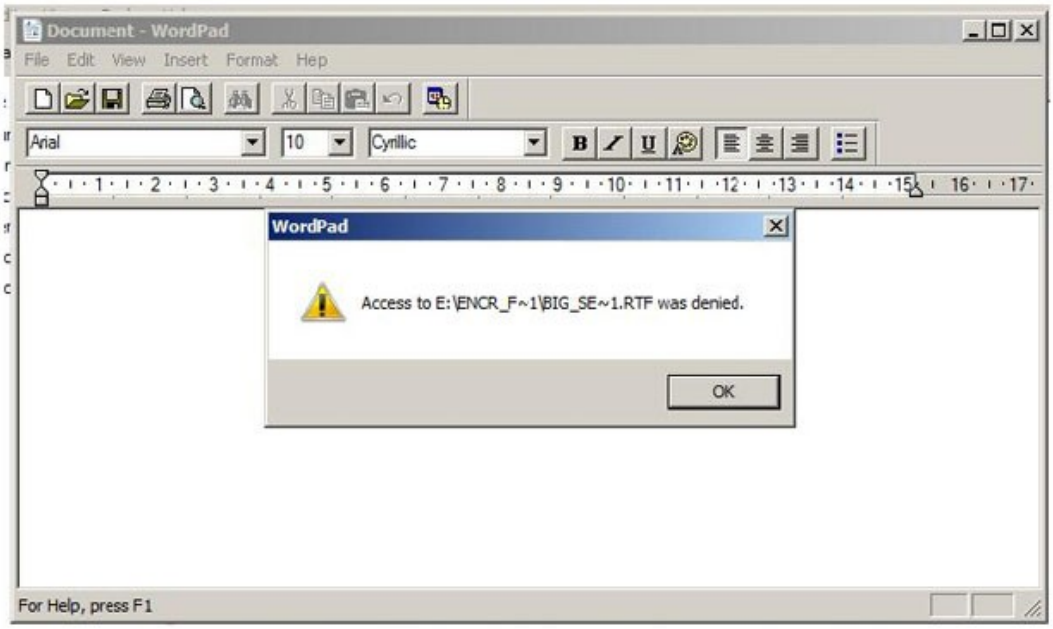


Рис. 7.2. Другой пользователь прочитать файл не сможет

Практическая часть

Задание 1

1. Работая под первой учетной записью, запросите сертификат (если он не был получен ранее), после чего зашифруйте папку с тестовым файлом, который не жалко потерять. Проверьте, что будет происходить при добавлении файлов, переименовании папки, копировании ее на другой диск с файловой системой NFTS на том же компьютере, копировании папки на сетевой диск или диск с FAT.

2. Убедитесь, что никто другой не сможет прочитать зашифрованный файл.

3. Снова зайдите под первой учетной записью. В оснастке **Certificates**, удалите *сертификат пользователя* (несмотря на выдаваемые системой предупреждения). Завершите сессию пользователя в системе и войдите заново. Попробуйте открыть зашифрованный файл.

Как вы убедились, если сертификат и соответствующая ему ключевая пара удалены, *пользователь* не сможет прочитать зашифрованные им же данные. В частности

поэтому, в *EFS* введена роль агента восстановления. Он может расшифровать зашифрованные другими пользователями данные.

Реализуется это примерно следующим образом. *Файл* шифруется с помощью симметричного криптоалгоритма на сгенерированном системой случайном ключе (назовем его **K1**). *Ключ* **K1** шифруется на открытом ключе пользователя, взятом из сертификата, и хранится вместе с зашифрованным файлом. Также хранится **K1**, зашифрованный на открытом ключе агента восстановления. Теперь либо *пользователь*, осуществлявший *шифрование*, либо *агент* восстановления могут *файл* расшифровать.

При настройке *по* умолчанию роль агента восстановления играет встроенная учетная *запись* администратора (локального, если *компьютер* не в домене, или доменная).

Задание 2

Зайдите в систему под встроенной учетной записью администратора и расшифруйте папку. То, какой *пользователь* является агентом восстановления, задается с помощью групповых политик. Запустим оснастку **Group Policy Management**. В политике домена найдем группу **Public Key Policies** и там **Encrypting File System**, где указан сертификат агента восстановления (рис. 7.3). Редактируя политику (*пункт* **Edit** в контекстном *меню*, далее **Policies —> Windows Settings —> Security Settings —> Public Key Policies —> Encrypting File System**), можно отказаться от присутствия агентов восстановления в системе или наоборот, указать более одного агента (рис. 7.4).

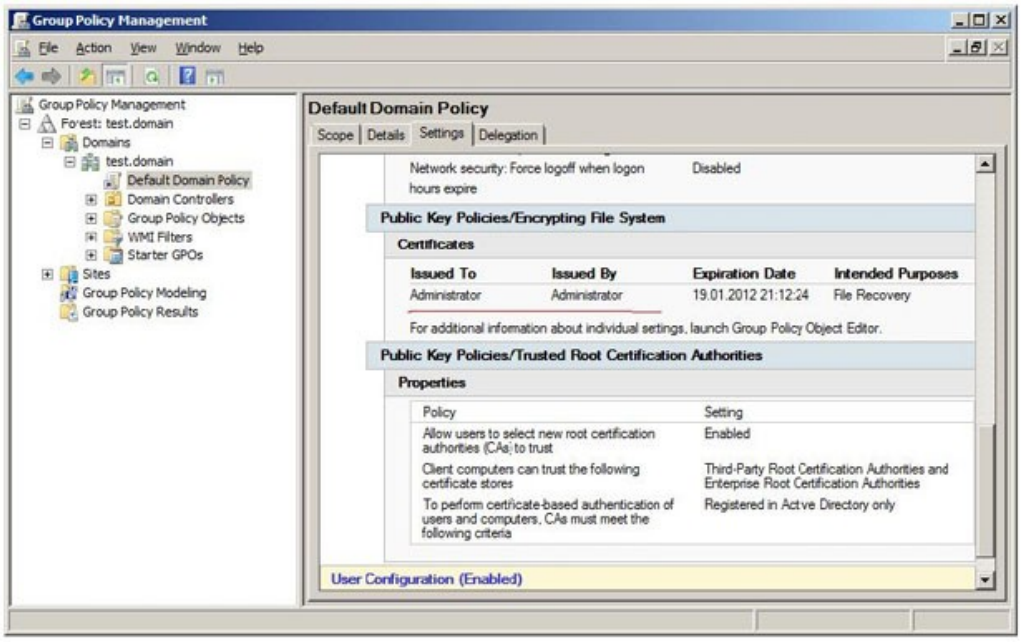


Рис. 7.3. Агент восстановления

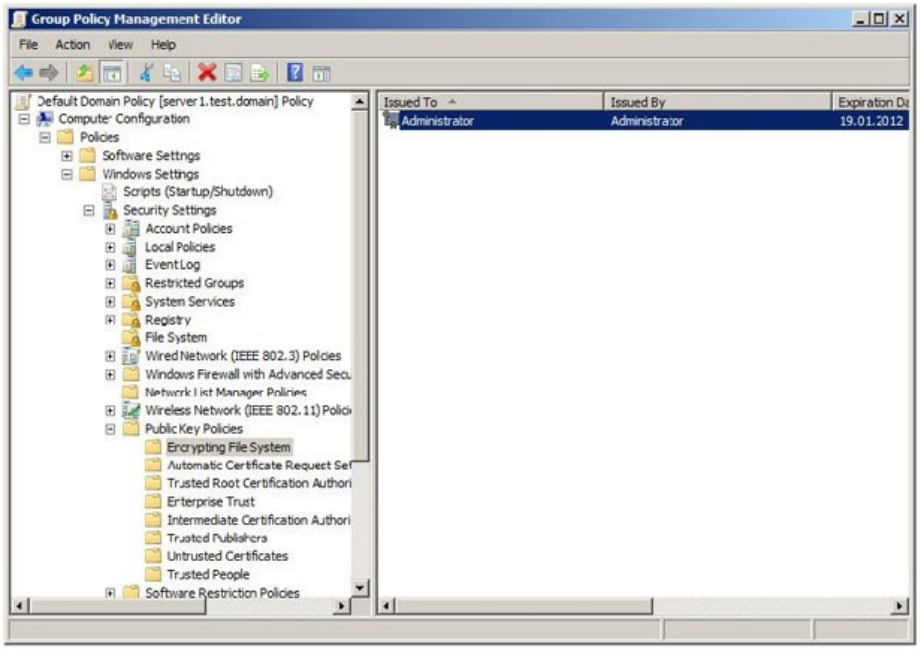


Рис. 7.4. Изменение агента восстановления

Задание 3

1. Отредактируйте политику таким образом, чтобы убрать из системы агента восстановления (удалите в *политике сертификат*). Выполнив команду "gpupdate /force" (меню **Start—>run—> gpupdate /force**) примените политику.

2. Повторив действия из предыдущих заданий, убедитесь, что теперь только тот пользователь, который зашифровал файл, может его расшифровать.

3. Теперь вернем в систему агента восстановления, но будем использовать новый сертификат. В редакторе политик находим политику **Encrypting File System** и в контекстном меню выбираем **Create Data Recovery Agent**. Это приведет к тому, что пользователь **Administrator** получит новый сертификат и с этого момента сможет восстанавливать шифруемые файлы.

Теперь рассмотрим, как можно предоставить *доступ* к зашифрованному файлу более чем одному пользователю. Такая настройка возможна, но делается она для каждого файла в отдельности.

В свойствах зашифрованного файла откроем окно с дополнительными параметрами, аналогичное представленному на рис. 7.1 для папки. Если нажать кнопку **Details**, будут выведены подробности относительно того, кто может получить *доступ* к файлу. На рис. 7.5 видно, что в данный момент это *пользователь* **User1** и *агент* восстановления **Administrator**. Нажав кнопку **Add** можно указать сертификаты других пользователей, которым предоставляется *доступ* к файлу.

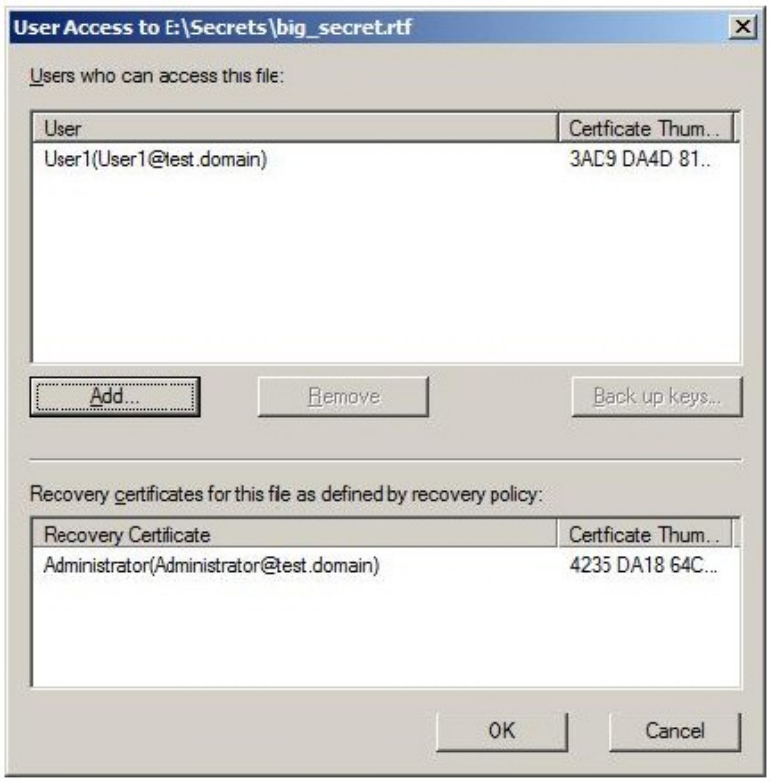


Рис. 7.5. Данные о пользователях, которые могут расшифровать файл

Задание 4

Зашифруйте *файл*. Предоставьте другому пользователю, не являющемуся агентом восстановления, возможность также расшифровать данный *файл*. Проверьте работу выполненных настроек.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: отчет.

8. Лабораторная работа № 8. Защита программ от НСИ с помощью USB-ключей.

Цель работы:

Ознакомиться с устройством USB ключей HASP. Программное обеспечение, поставляемое с устройством. Научиться защищать программы от не санкционированного использования с помощью программного обеспечения производителя ключей HASP.

1.Теоретическая часть

Проблема защиты программ

Пиратство в сфере информационных технологий и в части нелегального использования чужого программного обеспечения, становится доходным и динамично развивающимся полулегальным бизнесом с такими естественными составляющими, как производство, логистика, дистрибуция, прямые продажи и пр.

Приобретение и использование контрафактной (т. е. незаконной, ворованной) продукции, и тем более кража программных разработок и нажива на них, — это серьёзные преступления, которые наносят гигантский ущерб правообладателям. В связи с этим всё более актуальными становятся технологические вопросы защиты ПО.

Вопрос о защите программного обеспечения появился практически тогда же, когда появилось и оно само, и приобрёл особый вес с распространением персональных компьютеров. Всеобщая компьютеризация, помимо безусловных плюсов, таких как новые отрасли экономики, новые рынки, доступность богатейших библиотек, возможность общения, самореализации т. д., породила и ряд очевидных побочных эффектов, одним из которых является теневой бизнес по изготовлению и продаже нелицензионного программного обеспечения.

Незаконно растиражированные программы свободно продаются на рынках, и, покупая всё это, пользователь поощряет процветание и развитие пиратского бизнеса. Чего же лишается потребитель, покупая «нелицензионки»?

Гарантий качества продукта, технической документации и поддержки, бесплатных или льготных обновлений более совершенных версий ПО и ряда внешних атрибутов (фирменного оформления, вкладок и пр.). Впрочем, последнее, естественно, не может сформировать комплекс вины у обладателя пиратского диска. Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Пути и методы решения проблемы

Что на этом фоне остаётся делать программистам? Позаботиться о защите собственноручно написанных программ самим, не дожидаясь, пока представителей нелегального бизнеса вдруг заест совесть и они выйдут из тени.

На сегодняшний день есть несколько способов защиты своих программных продуктов от пиратов.

Подавляющее большинство создателей ПО используют различные программные модули, контролирующие доступ пользователей с помощью ключей активации, серийных номеров и т. д. Но невысокая стоимость — это, наверное, единственный плюс таких решений для борьбы со взломом программ. Интернет изобилует разного рода «крэками»1, патчами и эмуляторами, позволяющими обходить защиту и блокирующими запрос паролей или ключей активации. О серийных ключах говорить и вовсе не приходится. Легальный пользователь, купивший лицензионную программу, может просто обнародовать

код, по каким-то своим личным весьма альтруистическим соображениям, и не пройдёт и пары часов, как им воспользуется не одна сотня приверженцев всего бесплатного.

Эти очевидные недостатки привели к созданию аппаратной защиты программного обеспечения в виде электронного ключа. Первый ключ для защиты ПО на персональном компьютере был создан в начале 1980-х годов и с тех пор претерпел значительные изменения. Сегодня он представляет собой компактное USB-устройство (менее 4 см в длину), позволяющее шифровать данные, используя либо публичный, либо секретный алгоритмы.

Секретные алгоритмы разрабатываются самим производителем средств защиты, в том числе и индивидуально для каждого заказчика. Главным недостатком использования таких алгоритмов является невозможность оценки стойкости. С уверенностью сказать, насколько надёжен алгоритм, можно было лишь постфактум: взломали или нет. Кроме того, знание алгоритма даёт возможность создания эмулятора — программного продукта, полностью выполняющего все функции аппаратного устройства.

Публичный алгоритм, или «открытый исходник», обладает криптостойкостью несравнимо большей. Брюс Шнайер2 в своей статье «Открытые исходники и безопасность» объясняет преимущества публичных алгоритмов прежде всего тем, что они проверены не случайными людьми, а рядом экспертов, специализирующихся на анализе криптографии. «Прежде чем алгоритм будет признан надёжным, он должен быть изучен многими экспертами в течение ряда лет. И это — сильный довод в пользу криптографии с открытым кодом.

Поскольку единственный способ обрести уверенность в алгоритме — это предоставить его для изучения экспертам, а единственное условие, при котором они будут тратить время на проверку, — возможность публиковать результаты своих исследований, алгоритмы должны публиковаться», — пишет автор. Таким образом, становится очевидным, что «открытые исходники» разрабатываются с учётом того, что будут доступны всем желающим. Следовательно, риска их публикации нет.

Публичный криптоалгоритм кардинально усложняет задачу взлома ПО, сводя её к криптоанализу и вычислению ключей шифрования методом полного перебора. Самыми известными открытыми алгоритмами на сегодняшний день являются DES, Triple DES, Blowfish, RC2, CAST и, наверное, один из самых «продвинутых» на сегодняшний день, общепризнанный американский стандарт шифрования AES (Advanced Encryption Standard), пришедший на смену двум первым.

Для эффективного использования системы защиты HASP Вам следует ознакомиться с принципами её работы и терминологией, изложенными в настоящей главе. Если Вы собираетесь использовать ключ NetHASP.

Ключи HASP

Ключи HASP бывают следующих типов:

Локальные ключи – это ключи HASP, предназначенные для автономных (не являющихся частью сети) компьютеров. К этой категории относятся все ключи, кроме NetHASP. Демонстрационные ключи. В каждый комплект разработчика HASP входит демонстрационный ключ HASP (также демо-ключ). Демо-ключи обладают всеми возможностями ключей своего класса, но имеют стандартный демонстрационный код разработчика. Использовать демо-ключи для защиты программного обеспечения нельзя, поскольку они доступны для любого человека. Эти ключи лучше всего использовать для оценки системы защиты HASP.

Память HASP

Все ключи HASP, за исключением HASP4 без памяти, имеют перезаписываемую память. Используя память HASP, вы можете делать следующее: Управлять доступом к различным программным модулям и пакетам программ. Назначить каждому пользователю Ваших программ уникальный номер. Сдавать программы в аренду и распространять их

демо-версии с ограничением количества запусков. Хранить в ключе пароли, фрагменты кода программы, значения переменных и другую важную информацию.

Тип ключа Размер памяти
HASP4 без памяти Нет
HASP4 M1 112 байт
HASP4 M4 496 байт
HASP4 Time 512 байт
Все ключи NetHASP 496 байт
Идентификатор HASP

У каждого ключа HASP с памятью имеется уникальный опознавательный номер (ID-номер), или идентификатор, доступный для контроля защищёнными приложениями. Идентификаторы позволяют вам различать пользователей приложений. Проверяя в программе идентификатор HASP, Вы можете предпринимать те или иные действия в зависимости от наличия конкретного ключа. Вы не можете заказывать ключи HASP с заранее заданными идентификаторами. Они назначаются псевдослучайным образом в процессе изготовления ключей, чем гарантируется защита от повтора.

Способы защиты HASP

Система HASP позволяет защищать программное обеспечение двумя различными способами: Утилитой HASP Envelope (оболочка) HASP API (Application Programming Interface – программный интерфейс приложения) Оболочка HASP Использование HASP Envelope является основным способом защиты. Исполняемый файл заключается в защитную программную оболочку, кодирующую файл, и обладающую такими свойствами, как распознавание ключа и антиотладка. Оболочка не позволяет файлу выполняться без соответствующего ключа HASP.

Защита оболочкой производится быстро и без особых усилий. В то же время, она достаточно надёжна, так как делает отладку и дизассемблирование Ваших программ практически невозможными. Для защиты оболочкой исходные тексты программ не требуются.

Программный интерфейс пользователя HASP (API) Если у Вас имеются исходные тексты программы, которую надо защитить, то Вы можете пристыковать к ней модуль HASP API – объектный файл или библиотеку DLL.

Поскольку модуль API сам по себе защищён и зашифрован, этот метод обеспечивает высокую степень защиты. API позволяет обращаться к ключу.

8.2.Порядок выполнения работы

Установка программного обеспечения HASP

1. В списке оборудования компьютера убедиться в отсутствии устройств компании Aladdin (eToken,

HASP). Если эти устройства присутствуют в списке оборудования – удалить их.

2. Удалить, если присутствует, программное обеспечение eToken и HASP.

3. Подключиться к компьютеру Centurion, открыть ресурс Q и войти в каталог setup.

4. Запустить программу setup.exe.

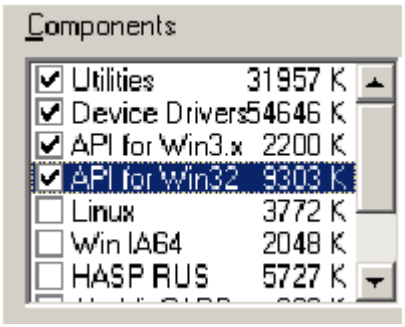


Рис. 8.1

6. Следуя указаниям программы установки установить на компьютере программное обеспечение системы HASP. Установить компоненты помеченные ниже.

6. Подключить HASP к разъему USB компьютера.

a) В случае правильной установки в электронном ключе HASP должен загореться световой индикатор.

b) Если индикатор не загорается, обновить драйвер устройства вручную.

7. В списке оборудования компьютера найти подключенное устройство.

Работа с программным обеспечением HASP

1.

Запустить программу Aladdin DiagnostiX провести тестирование электронного ключа HASP.

2.

Запустить демонстрационную программу – HASP Demo for Win16 . Описать работу этой программы.

3.

Запустить демонстрационную программу – HASP Demo for Win32 . Описать работу этой программы.

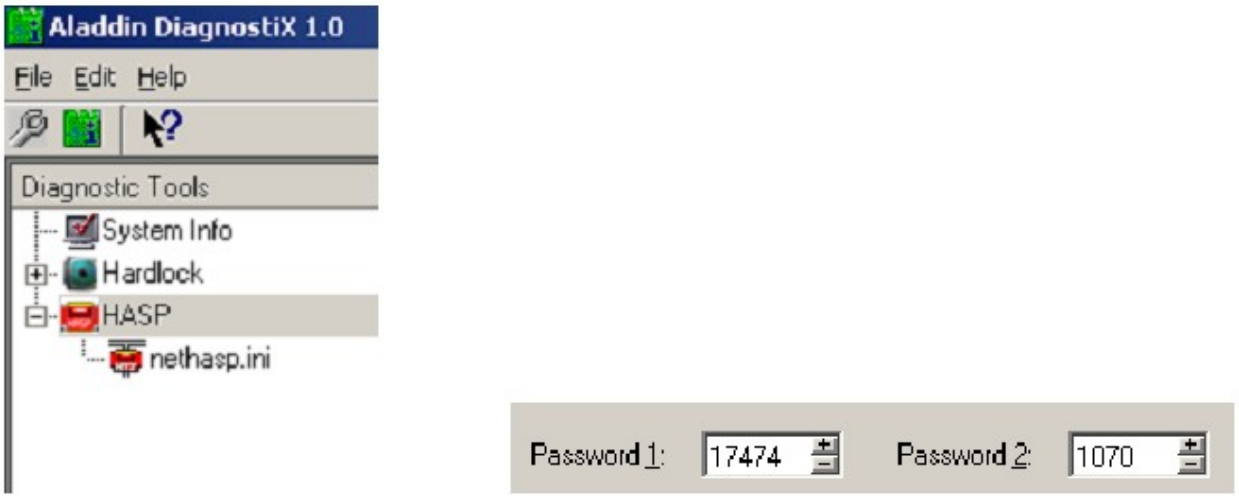


Рис. 8.2.

Зашита приложений с помощью HASP.

Используя инструмент HASP Envelope защитить избранное приложение от не санкционированного запуска.

Санкционированный запуск – при наличии HASP ключа в системе.

Несанкционированный запуск – при отсутствии HASP ключа в системе.

1. Создать папку: «Защищенные приложения»
2. Скопировать в нее 3 исполняемых файла, например программу калькулятор – calc.exe
3. С помощью HASP Envelope защитить выбранные программы от несанкционированного запуска.
4. Запустить программу.
5. Закрыть программу
6. Вытащить HASP из разъема USB
7. Запустить защищенную программу.

Попробовать варианты защиты нескольких программ с одним ключом HASP.

8.3. Содержание отчета

Форма отчётности - подробное описания проделанной работы по каждому шагу в электронном документе (Microsoft Word). Привести структуру ключевой дискеты. Содержание файла открытых и закрытых ключей.

8.4. Контрольные вопросы

1. Рассказать об утилите HASP Envelope
2. Рассказать об утилите тестирования HASP
3. Рассказать об утилите HASPEdit
4. Что такой идентификатор HASP
5. Как шифруются и дешифруются данные для распознавания ключа HASP.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: отчет.

9.Лабораторная работа № 9. Sandbox – файловые антивирусы.

Цель работы:

Получение теоретических и практических навыков работы с песочницами и файловыми антивирусами.

9.1.Теоретическая часть

Вредоносная программа (англ. malware, malicious software — «злонамеренное программное обеспечение») — любое программное обеспечение, предназначенное для получения несанкционированного доступа к вычислительным ресурсам самой ЭВМ или к информации, хранимой на ЭВМ, с целью несанкционированного использования ресурсов ЭВМ или причинения вреда (нанесения ущерба) владельцу информации, и/или владельцу ЭВМ, и/или владельцу сети ЭВМ, путем копирования, искажения, удаления или подмены информации. Многие антивирусы считают крэки, кейгены и прочие программы для взлома вредоносными программами, или потенциально опасными.

Песочница (англ. Sandbox) - это ограниченная среда в вашей системе для исполнения гостевых программ без доступа к главной операционной системе. Это закрытое для доступа извне виртуальное пространство, в котором можно работать с программным обеспечением без изменения системных файлов.

Песочница (от англ. Sandbox, схожие понятия — англ. honeypot, англ. fishbowl) - в компьютерной безопасности, механизм для безопасного исполнения программ. Песочницы часто используют для запуска непротестированного кода, непроверенного кода из неизвестных источников, а также для запуска и обнаружения вирусов.

Песочница — в компьютерной безопасности специально выделенная среда для безопасного исполнения компьютерных программ.

Песочница обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность сообщаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно ограничивают. Песочницы представляют собой пример виртуализации. Повышенная безопасность исполнения кода в песочнице зачастую связана с большой нагрузкой на систему — именно поэтому некоторые виды песочниц используют только для неотлаженного или подозрительного кода.

Cuckoo Sandbox — система для автоматического исследования вредоносного ПО, эксплоитов, вредоносных скриптов, документов, архивов и ссылок. Система способна проверять документы pdf, doc, xls, rtf, скрипты Python, JS, DLL библиотеки, бинарники, jar и многое другое.

Файловый Антивирус — компонент модуля Защита компьютера, контролирующий файловую систему компьютера. Он запускается при старте операционной системы, постоянно находится в оперативной памяти компьютера и проверяет все открываемые, сохраняемые и запускаемые файлы. Каждый файл, к которому вы обратитесь, будет перехвачен Файловым Антивирусом и проверен на присутствие известных вирусов.

Clam AntiVirus — пакет антивирусного ПО, работающий во многих операционных системах, включая Unix-подобные ОС, OpenVMS, Microsoft Windows и Apple Mac OS X.

Главная цель Clam AntiVirus — интеграция с серверами электронной почты для проверки файлов, прикреплённых к сообщениям. В пакет входит масштабируемый многопоточный демон clamd, управляемый из командной строки сканер clamscan, а также модуль обновления сигнатур по Интернету freshclam.

Задания к лабораторной работе

Установите Cukoo Sandbox.

Примечание

Если будут проблемы с установкой, воспользуйтесь документацией Cukoo sandboxhttp://docs.cuckoosandbox.org/en/latest/installation/

* Установка зависимостей

```
cd /tmp
apt-get update
apt-get install  git automake mongodb mingw32 dkms unzip wget python python-sqlalchemy python-bson python-pip python-dpkt python-jinja2 python-magic python-mysqldb python-gridfs python-libvirt python-bottle python-pefile python-chardet -y

apt-get install python-dev libxml2-dev libxslt1-dev libevent-dev libpcre3 libpcre3-dev zlib1g-dev libtool libpcre++-dev –y
apt-get install mariadb-server -y
pip install lxml
pip install cybox==2.0.1.4
pip install maec==4.0.1.0
pip install django
pip install py3compat
pip install pymongo
apt-get install ssdeep python-pyrex subversion libfuzzy-dev -y
svn checkout http://pyssdeep.googlecode.com/svn/trunk/ pyssdeep
cd pyssdeep
python setup.py build
python setup.py install
pip install pydeep
cd /tmp
wget https://github.com/plusvic/yara/archive/v2.1.0.tar.gz
tar xzf v2.1.0.tar.gz
cd yara-2.1.0
chmod +x build.sh
./build.sh
make install
cd yara-python
python setup.py build
python setup.py install
cd /tmp
wget http://distorm.googlecode.com/files/distorm3.zip
unzip distorm3.zip
cd distorm3
python setup.py build
python setup.py install
add-apt-repository ppa:pi-rho/security
apt-get update
apt-get install volatility
```

Установка и настройка Virtualbox:

```
wget -q http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add – sh -c 'echo "deb http://download.virtualbox.org/virtualbox/debian trusty contrib" >> /etc/apt/sources.list.d/virtualbox.list'
```

```
apt-get update
apt-get install virtualbox-4.3
cd /tmp
VBOX_LATEST_VERSION=$(curl
http://download.virtualbox.org/virtualbox/LATEST.TXT)
wget
http://download.virtualbox.org/virtualbox/${VBOX_LATEST_VERSION}/Oracle_VM_Virtual
Box_Extension_Pack-{VBOX_LATEST_VERSION}.vbox-extpack
vboxmanage extpack install /tmp/Oracle_VM_VirtualBox_Extension_Pack-$
{VBOX_LATEST_VERSION}.vbox-extpack
cd /opt
wget http://dlc.sun.com.edgesuite.net/virtualbox/${VBOX_LATEST_VERSION}/
VBoxGuestAdditions_${VBOX_LATEST_VERSION}.iso
```

Установка Cuckoo Sandbox:

```
useradd cuckoo
usermod -a -G vboxusers cuckoo
id cuckoo
cd /opt
wget http://downloads.cuckoosandbox.org/1.1/cuckoo_1.1.tar.gz
tar xzf cuckoo_1.1.tar.gz
```

Настройка Cuckoo Sandbox:

```
cd /opt/cuckoo
./utils/community.py --signatures --force
```

Настройка БД:

```
mysql -u root -p
> create database cuckoo;
> grant all privileges on cuckoo.* to cuckoo@localhost identified by 'cuck00pass' ;
> flush privileges;
> quit;
```

Настраиваем cuckoo::

* Файл /opt/cuckoo/conf/cuckoo.conf

Включаем запись дампа памяти:

memory_dump = on

Настраиваем подключение к бд:

connection = mysql://cuckoo:cuck00pass\@localhost/cuckoo

Увеличиваем временные лимиты:

default = 240

critical = 1200

vm_state = 600

* Файл /opt/cuckoo/conf/memory.conf

Отключаем сохранение дампов памяти:

delete_memdump = yes

* Файл /opt/cuckoo/conf/processing.conf

Включаем анализ оперативной памяти:

memory = yes

* nano /opt/cuckoo/conf/virtualbox.conf

Меняем режим работы Virtualbox:

mode = headless

Меняем названия виртуальной машины с cuckoo1 на WindowsXP:
machines = WindowsXP
[WindowsXP]
label = WindowsXP
* Файл /opt/cuckoo/conf/reporting.conf
Включим импорт отчётов в MongoDB для работы веб интерфейса
[mongodb]
enabled = yes

Примечание

• На этом настройка Cuckoo закончена, теперь приступим к Virtualbox и гостевой ОС.

Загрузка виртуальной ОС с сайта:
wget https://az412801.vo.msecnd.net/vhd/VMBuild_20131127/VirtualBox/IE6_WinXP/Linux/IE6.WinXP.For.LinuxVirtualBox.sfx
chmod +x IE6.WinXP.For.LinuxVirtualBox.sfx
./IE6.WinXP.For.LinuxVirtualBox.sfx
vboxmanage import IE6\ -\ WinXP.ova --vsys 0 --unit 10 --disk=/root/VirtualBox\ VMs/WindowsXP/WindowsXP.vmdk --memory 1024 --vmname WindowsXP
Настраиваем сеть:
vboxmanage hostonlyif create

vboxmanage modifyvm "WindowsXP" --nic1 hostonly --hostonlyadapter1 vboxnet0 --nicpromisc1 allow-all --hwvirtex off --vtxvpid off
Настраиваем общие папки:
mkdir -p /opt/cuckoo/shares/setup

mkdir -p /opt/cuckoo/shares/WindowsXP

vboxmanage sharedfolder add "WindowsXP" --name "WindowsXP" --hostpath /opt/cuckoo/shares/WindowsXP --automount

vboxmanage sharedfolder add "WindowsXP" --name setup --hostpath /opt/cuckoo/shares/setup --automount --readonly

vboxmanage modifyvm "WindowsXP" --nictrace1 on --nictracefile1 /opt/cuckoo/shares/WindowsXP/dump.pcap
Включаем доступ по RDP, порт можете указать любой:
vboxmanage modifyvm "WindowsXP" --vrdeport 5000 --vrde on

Примечание

На этом конфигурация виртуальных контейнеров полностью закончена, осталось настроить iptables, tcpdump.:
iptables -A FORWARD -o eth0 -i vboxnet0 -s 192.168.56.0/24 -m conntrack --ctstate NEW -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A POSTROUTING -t nat -j MASQUERADE
sysctl -w net.ipv4.ip_forward=1
setcap cap_net_raw,cap_net_admin=eip /usr/sbin/tcpdump
getcap /usr/sbin/tcpdump

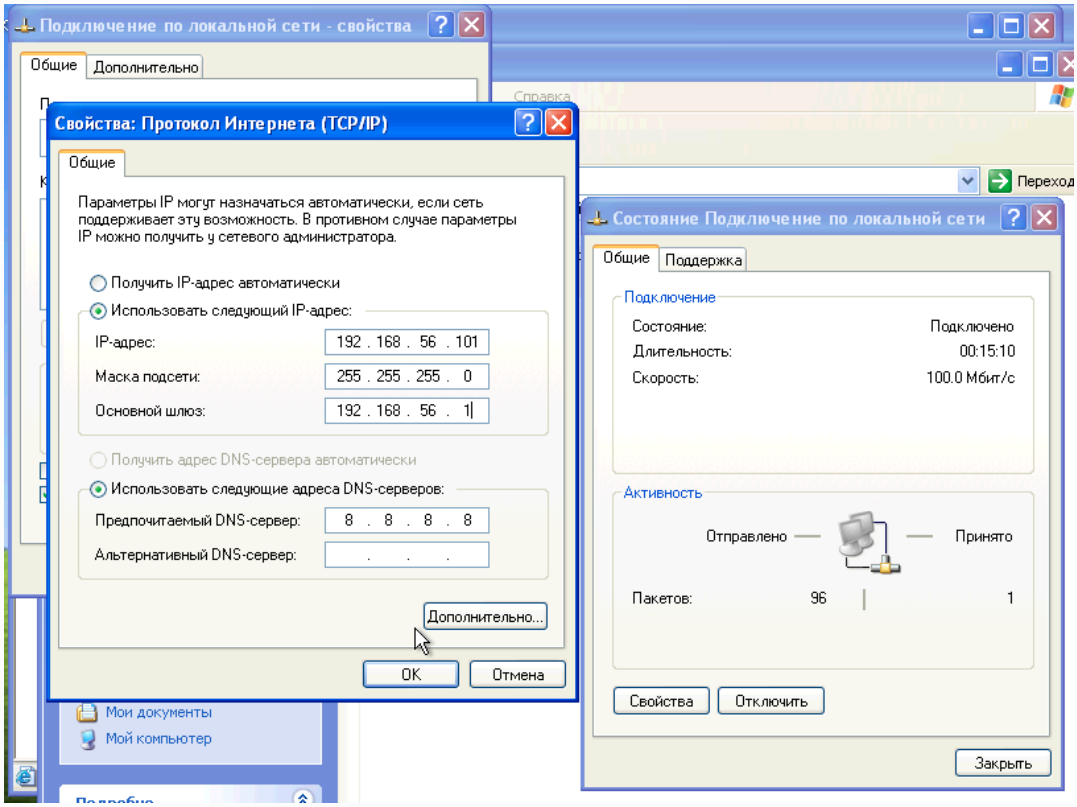
ifconfig vboxnet0 192.168.56.1

- После установки и настройки Windows переходим непосредственно на саму гостевую ОС.

Предупреждение

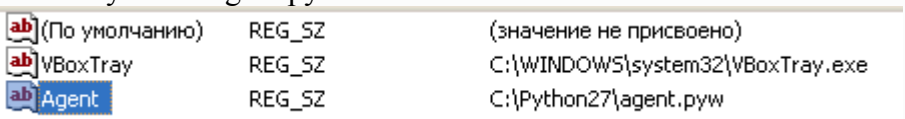
(Если возникли проблемы с установкой виртуальной машины, можете посмотреть, как это сделано в этой статье http://gwallgofi.com/cuckoo-sandbox-part-2-installing/)

- Следующим образом настроим подключение к сети (dns можете указать любой):



- Установим VboxTools с диска, который подключен к системе.
- Устанавливаем Pyton 2.7: http://python.org/download/
- Устанавливаем http://www.activestate.com/activepython.
- Устанавливаем PIL Python модуль, для создания скриншотов: http://www.pythonware.com/products/pil/.
- Отключаем автоматическое обновление Windows.
- Отключаем брандмауэр.
- Копируем агент из сетевой папки setup в папку C:Python27, Ставим агент на автозагрузку, для этого добавляем в ветку реестра(пуск->выполнить->regedit) HKLMSOFTWAREMicrosoftWindowsCurrentVersionRun строковый параметр

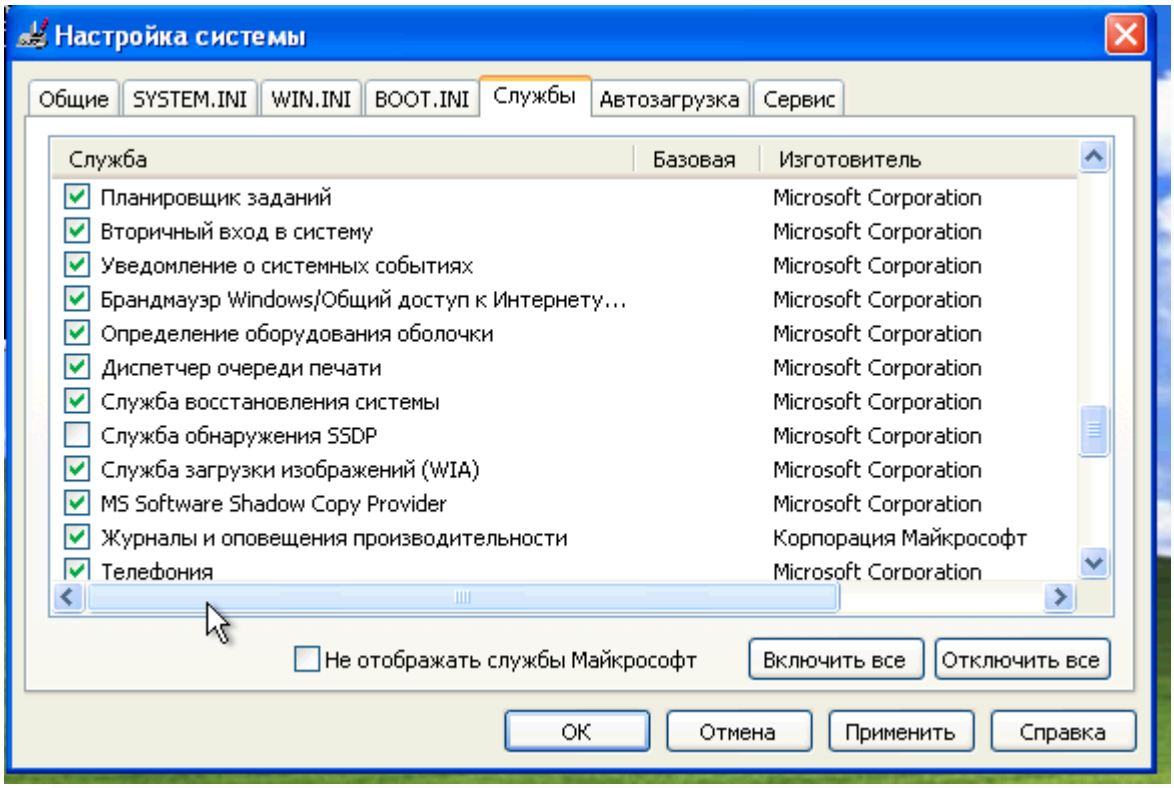
Имя:'Agent'
Тип:'REG_SZ'
Содержание:«C:Python27agent.pyw»



• Включаем IE, в настройках ставим домашней страницей пустую вкладку, по желанию в свойствах обозревателя выключаем все защитные механизмы.

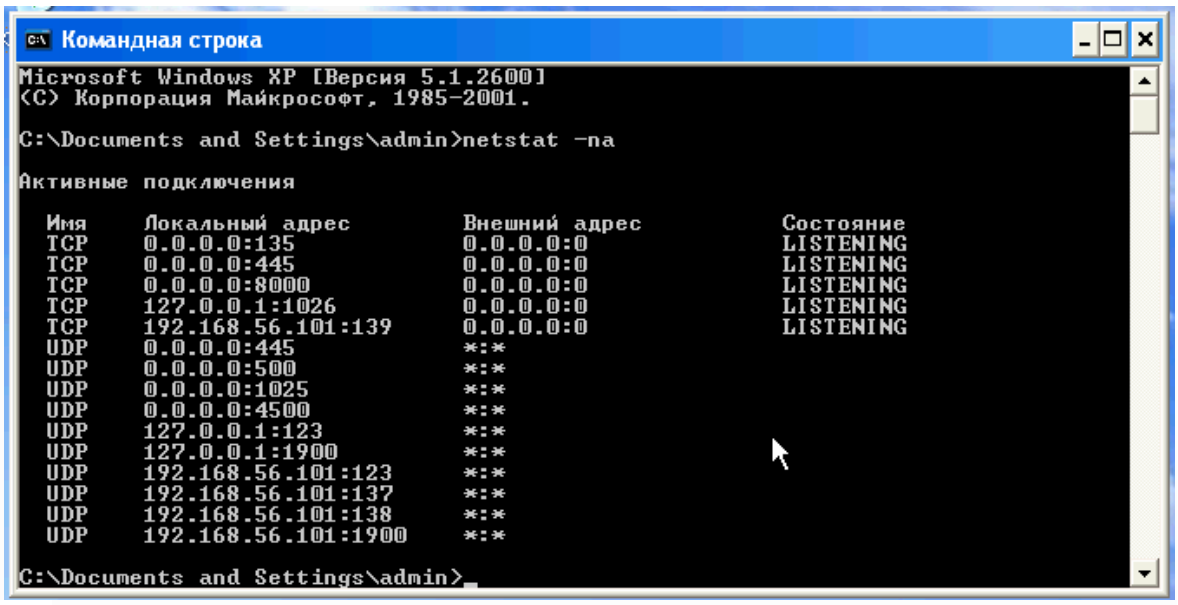
• Отключаем SSDP: пуск->выполнить->msconfig и в разделе службы отключаем «Служба обнаружения SSDP», чтобы в отчётах не фигурировали сетевые запросы этой службы.

•



• Перезагружаемся и в появившемся при загрузке окне выбираем «При перезагрузке не выводить это сообщение» и ОК.

• После перезагрузки гостевой ОС, пуск->выполнить->cmd и в консоли набираем netstat –na и смотрим есть ли агент на 8000-ом порту



- По желанию устанавливаем различное уязвимое ПО старых версий (браузеры, Flash player, Java, Acrobat Reader...)

- На этом установка гостевой ОС закончена. Делаем снапшот (не выключая гостевую ОС) <vboxmanage snapshot “WindowsXP” take “WindowsXPSnap01” –pause>

- И выключаем: <vboxmanage controlvm “WindowsXP” poweroff>

- Запуск Cukoo Sandbox <python cuckoo.py>

- Загрузите вредоносное ПО, для проверки работы Cukoo Sandbox. Здесь есть небольшой список ресурсов с образцами вредоносного ПО https://zeltser.com/malware-sample-sources/. Например, Можете загрузить с http://malshare.com/ или с https://malwr.com/ .

- Чтобы отправить файл в Cukoo на анализ используйте команду submit: <python submit.py /path/to/binary>

- Запустите web-интерфейс cukoo и просмотрите результаты <python web.py>

Вопросы к лабораторной работе

1. Что такое песочница?
2. Принцип работы песочниц.
3. Где используют песочницы?
4. Преимущества и недостатки песочниц.
5. Альтернативы использованию песочниц.
6. Что такое эмуляция?
7. Что такое эвристический анализ? В чем отличия от сигнатурного анализа?
8. Для чего нужен файловый антивирус?
9. Что такое вредоносное ПО?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1	1-2	1-3	1-3

Оценочные средства: Собеседование, отчет.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

3.1. Рекомендуемая литература

3.1.1. Основная литература:

1. Царев, Р.Ю. Программные и аппаратные средства информатики: учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670
2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

3.1.2. Дополнительная литература:

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412
2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

1.3. Методическая литература:

1. Методические указания по выполнению лабораторных работ по дисциплине «Программно-аппаратные средства защиты информации».
2. Методические указания по выполнению практических работ по дисциплине «Программно-аппаратные средства защиты информации».
3. Методические рекомендации для студентов по организации самостоятельной работы по дисциплине «Программно-аппаратные средства защиты информации»

3.1.4. Интернет-ресурсы:

http://www.biblioclub.ru/ - электронная библиотека
http://www.uts-edu.ru/ - «Электронные курсы»

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

К практическим занятиям
по дисциплине
«ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ»
для направления подготовки **10.03.01 Информационная безопасность**
направленность (профиль) **Безопасность компьютерных систем**

Пятигорск
2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Цель и задачи изучения дисциплины

Цель дисциплины: формирование набора универсальных и общепрофессиональных компетенций будущего бакалавра (специалиста) по направлению подготовки 10.03.01 Информационная безопасность.

Задачами дисциплины являются: дать основы о методах и средствах защиты информации в компьютерных системах; дать основы правил разграничения доступа и основных функций СЗИ, его обеспечивающих; дать основы практических аспектов построения систем ограничения доступа и других СЗИ; дать основы аппаратной реализации различных средств защиты информации; дать основы о защитных механизмах, реализованных в средствах защиты компьютерных систем от несанкционированного доступа (НСД); дать основы вопросов защиты ПО от несанкционированного использования; дать основы о применении средств криптографической защиты информации и средств защиты от НСД для решения задач обеспечения информационной безопасности; дать основы методов защиты от РПВ; дать основы методов и особенностей защиты объектов ОС; дать основы принципов построения файловой системы и моделей разграничения доступа к объектам.

2. Оборудование и материалы

Для проведения практических занятий необходимо следующее материально-техническое обеспечение: персональный компьютер; проектор; возможность выхода в сеть Интернет для поиска по образовательным сайтам и порталам; интерактивная доска.

3. Наименование практических занятий

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов
5 семестр		
1	**Практическое занятие 1.** Сбор данных об информационной системе с помощью средств администрирования Windows.	6.75
1	**Практическое занятие 2.** Сбор данных о топологии сети с помощью средства администрирования сетей.	6.75
5	**Практическое занятие 3.** Идентификация и аутентификация систем семейства Microsoft Windows.	6.75
5	**Практическое занятие 4.** Аутентификация по протоколу Kerberos.	6.75
	Итого за 5 семестр	27
6 семестр		
19	**Практическое занятие 5.** Настройка локальной политики парольной безопасности операционной системы.	3
19	**Практическое занятие 6.** Инфраструктура открытых ключей. Цифровые сертификаты.	3
20	**Практическое занятие 7.** Использование цифровых сертификатов.	3
20	**Практическое занятие 8.** Резервное копирование в Windows Server 2012.	3
	Итого за 6 семестр	12
	Итого	**39**

4. Содержание практических занятий

3.1.ПРАКТИЧЕСКАЯ РАБОТА 1. Сбор данных об информационной системе с помощью средств администрирования Windows.

Цель работы:

Целью данной лабораторной работы является сбор данных об имеющихся компьютерах, установленных на них операционных системах, предоставляемых в общий *доступ* файловых ресурсах.

Содержание:

Получение перечня компьютеров домена. Определение перечня общих ресурсов рабочей станции. Управление компьютером. Информация о членстве пользователя в доменных группах. Разрешения файловой системы.

Теоретические основы.

Для проведения оценки рисков необходимо провести инвентаризацию активов информационной системы (ИС). Если в ИС используются домены *Windows*, для получения данных о системе можно использовать средства администрирования, реализованные в виде оснасток консоли администрирования (Microsoft *management* console - *mmc*).

Используемые в данной работе инструменты могут быть запущены из раздела **"Администрирование"** *меню* **"Пуск"** или через **"Панель управления" (Пуск Панель управления Администрирование)**.

Целью данной лабораторной работы является сбор данных об имеющихся компьютерах, установленных на них операционных системах, предоставляемых в общий *доступ* файловых ресурсах.

Из раздела **"Администрирование"** запустите **Active Directory Users and Computers**. В раскрывающемся списке объектов выберите **Ваш домен**, там откройте перечень компьютеров (*папка* **Computers** - рис. 1.1).

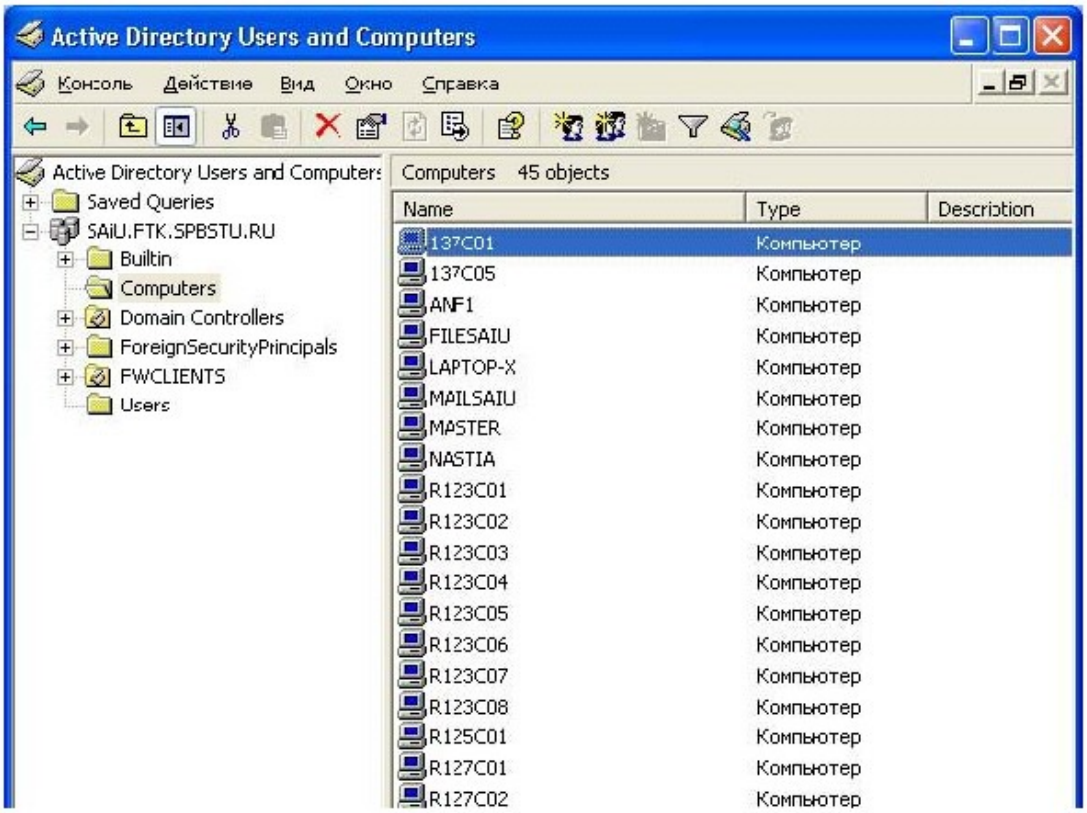


Рис. 1.1. Получение перечня компьютеров домена

С помощью кнопки панели инструментов **"Экспорт списка"** (на кнопке изображение списка и стрелки) *список* компьютеров можно экспортировать в *текстовый файл* для дальнейшей обработки. В свойствах компьютера отображается название и версия установленной операционной системы (рис. 1.2). Также там может быть дополнительная *информация*, например, описывающая *размещение*.

Аналогичные данные о контроллерах домена можно получить в разделе **Domain Controllers**. Данные о пользователях и их группах доступны в разделе **Users**. Надо отметить, что представленное распределение *по* разделам не является обязательным. В процессе администрирования могут создаваться новые *подразделения* (OU - Organization Unit) и объекты (например, пользователи или компьютеры) - помещаться в них.

Информацию о соответствии имен компьютеров IP-адресам можно получить, используя *утилиту командной строки* **nslookup** или административную оснастку **"DNS"**. Например, узнать *IP-адрес* компьютера http://comp1.mcompany.ru можно с помощью команды nslookup comp1.mcompany.ru Часто действующие настройки в сети таковы, что ip-адреса компьютерам выделяются динамически, с использованием службы **dhcp**, и могут периодически меняться. Как правило, у серверов ip-адреса постоянны.

Теперь перейдем к этапу сбора данных об информационных ресурсах, поддерживаемых на компьютере. Перечень предоставляемых в общий *доступ* папок можно получить с помощью оснастки **"Управление компьютером"**. На рис. 1.3 представлен пример перечня ресурсов рабочей станции, предоставляемых в общий *доступ* в служебных целях. Этот *список* можно также экспортировать в *текстовый файл*.

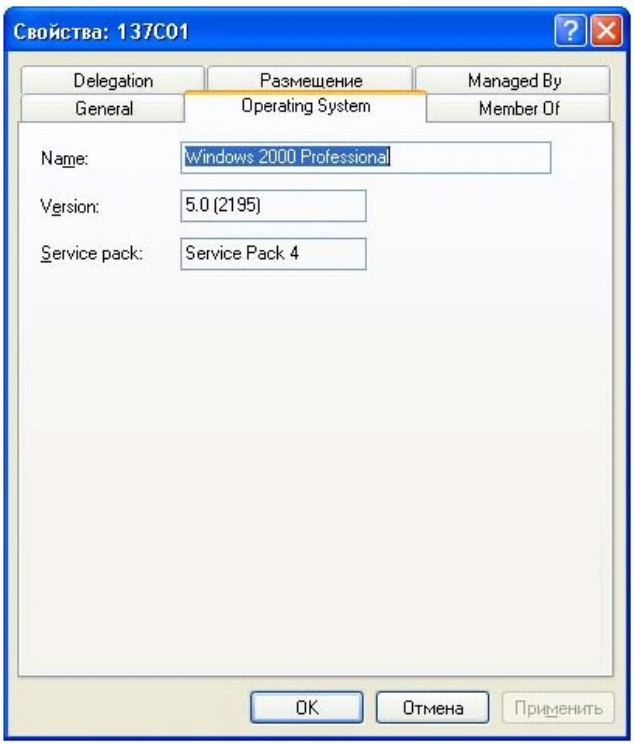


Рис. 1.2. Информация о компьютере

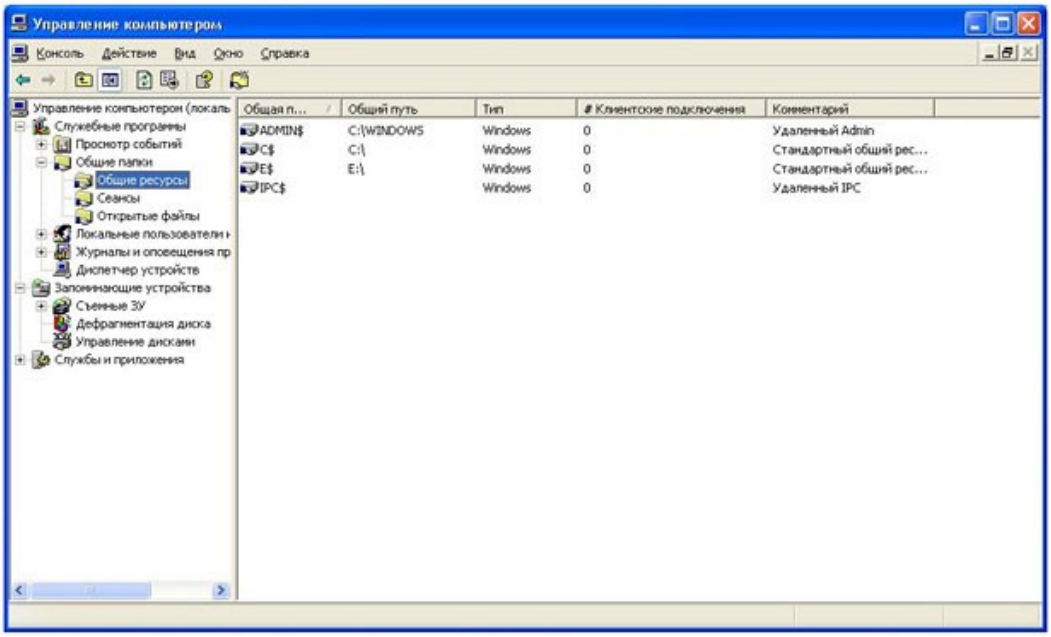


Рис. 1.3. Пример перечня общих ресурсов рабочей станции

Более интересен будет подобный *список* для файлового сервера. Чтобы его увидеть, надо подключить оснастку **"Управление компьютером"** для сервера. Запустите *консоль MMC* (**Пуск**→**Выполнить**→**mmc**), в *меню* выберите добавление новой оснастки (рис. 1.4), выберите оснастку **"Управление компьютером"** и укажите, что она будет использоваться для другого компьютера (рис. 1.5).

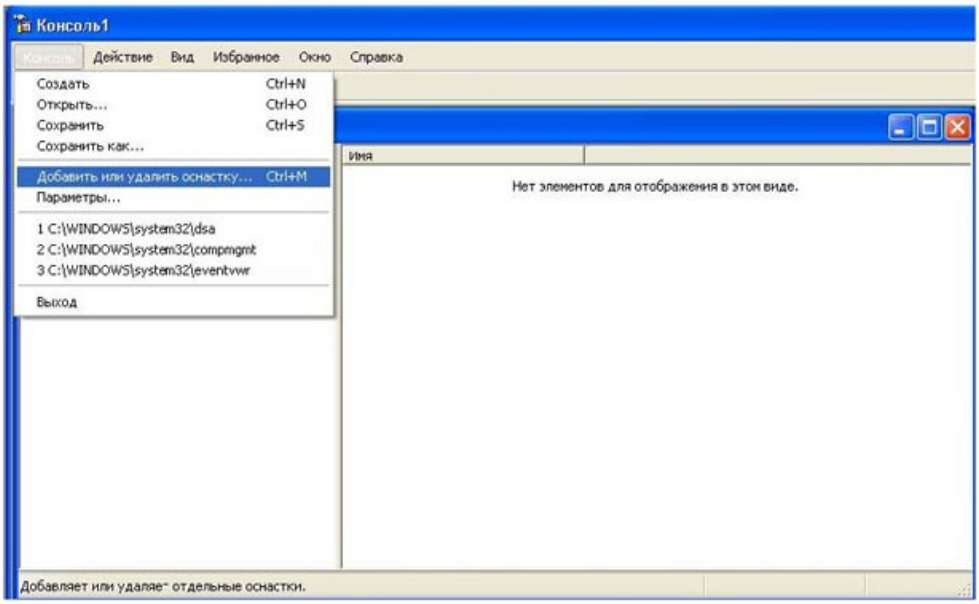


Рис. 1.4. Добавление новой оснастки

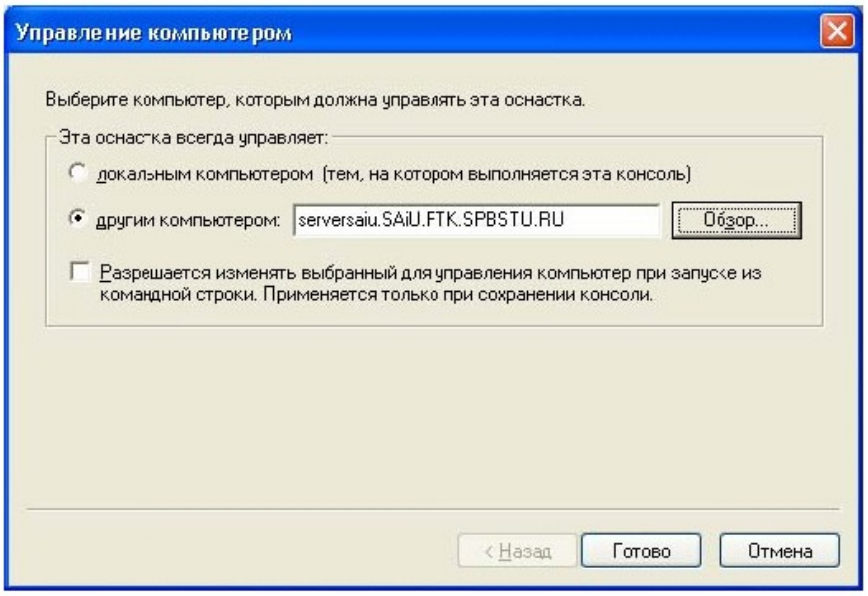


Рис. 1.5. Выбор компьютера

В остальном для пользователя все будет происходить так же, как и при работе с локальным компьютером.

В свойствах ресурса можно узнать о разрешениях, которые установлены на него как для разделяемого ресурса (рис. 1.6), а на вкладке **"Безопасность"** - разрешениях файловой системы *NTFS* (если *папка* расположена на разделе с этой файловой системой, а не с *FAT*).

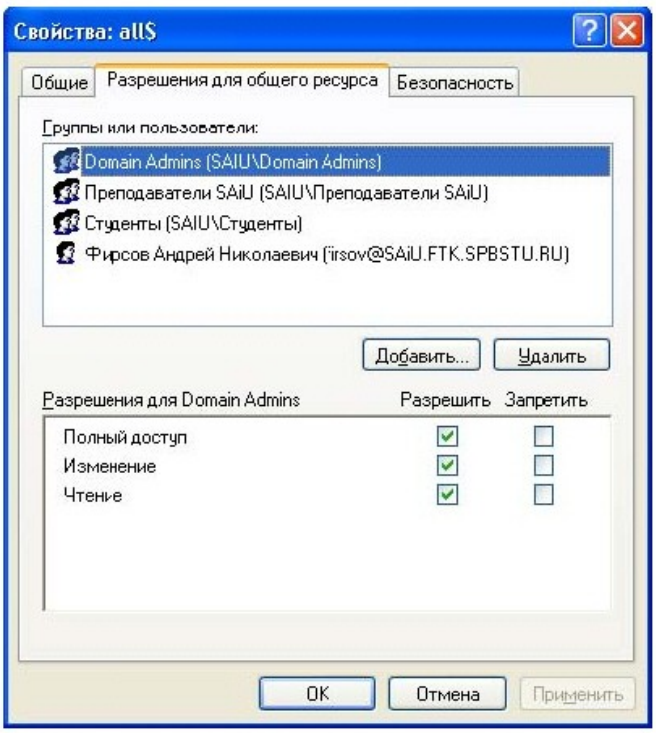


Рис. 1.6. Разрешения

Задание

1. Получите перечень компьютеров и контроллеров домена. Для указанных преподавателем 1-2 компьютеров выясните установленную операционную систему и используемые ими ip-адреса. Занесите данные в отчет.

2. Получите перечень предоставляемых в общий доступ каталогов на вашем компьютере и на компьютерах, данные о которых Вы собирали на этапе 1. Опишите хранимые там данные и охарактеризуйте степень их важности. Занесите полученную информацию в отчет.

3. Для указанных ресурсов и выбранных пользователей опишите действующие разрешения на доступ. При этом надо учитывать, что:

- эффективное (действующее) разрешение складывается из разрешений для пользователя лично и разрешений всех групп, в которые пользователь входит;
- запрещение имеет больший приоритет, чем разрешение;
- при комбинации разрешений для общего ресурса с разрешениями NTFS, приоритетными будут разрешения, максимально ограничивающие доступ.

Информацию о членстве пользователя в доменных группах можно получить через оснастку **Active Directory Users and Computers**, о локальных группах – через **"Управление компьютером"**

Контрольные вопросы:

1. Линии связей локальных сетей.
2. Обеспечение секретности передаваемой информации в локальных сетях.
3. Типы линий связи локальных сетей.
4. Бескабельные каналы связи локальных сетей.

5. Базовые технологии локальных сетей.
6. Нестандартные топологии локальных сетей.
7. Согласование и экранирование линий связи.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.2.ПРАКТИЧЕСКАЯ РАБОТА 2. Сбор данных о топологии сети с помощью средства администрирования сетей.

Цель работы:
Изучение различных вариантов сбора данных о топологии сети с помощью средств администрирования сетей с помощью *3Com Network Supervisor* .
Содержание:
Выбор документируемой сети. Функции, доступные для выбранного узла. Соединения по типам подключений. Данные о подключениях и графики. Свойства коммутатора. Средства удаленного администрирования.

Продолжая тему инвентаризации активов информационной системы (ИС), перейдем к рассмотрению средств, позволяющих получить данные о составе и топологии сети. В качестве примера в данной лабораторной работе будет использоваться *утилита 3Com Network Supervisor*, которую можно бесплатно получить с сайта компании *3Com* (http://www.3com.com). Аналогичные *по* функциональности продукты есть и у других производителей сетевого оборудования.

При запуске программы предлагается выбор - строить новую карту сети или открыть существующую. При выборе создания новой карты надо указать, какая *подсеть* документируется (рис. 2.1). На рисунке выбрана локальная *подсеть*, т.е. та ip-*сеть*, к которой относится *компьютер*, на котором выполняется *3Com Network Supervisor*.

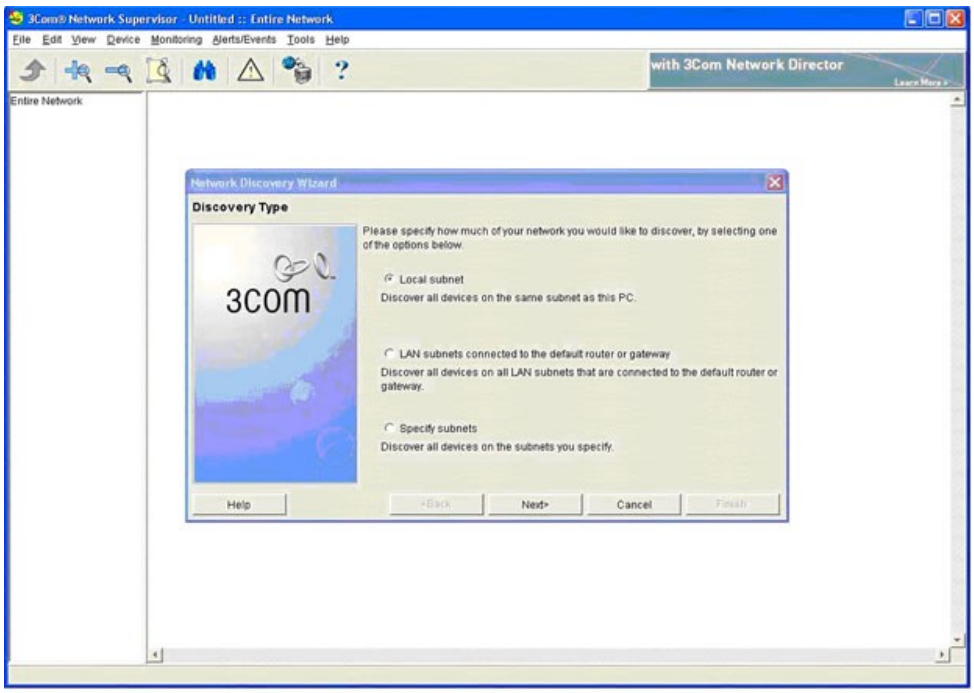


Рис. 2.1. Выбор документируемой сети

На рис. 2.2 представлен пример карты сети, которую строит *утилита*. Надо отметить, что наиболее информативна такая карта будет в том случае, если в сети используется управляемое сетевое оборудование *3Com*, поддерживающее, в частности, протокол *SNMP*. В то же время, польза от составления карты будет и в случае отсутствия в сети подобного оборудования.

Для того, чтобы это продемонстрировать, были сделаны следующие настройки. Каждому из компьютеров были присвоены ip-адреса из двух сетей класса С - 192.168.1.0 и 192.168.100.0. Управляемому коммутатору *3Com* SuperStack II *Switch* 3000 назначен

адрес 192.168.100.6, т.е. он "виден" только при построении карты сети 192.168.100.0. *DNS* серверы доступны только в сети 192.168.1.0, поэтому на рисунках, относящихся ко второй сети, компьютеры идентифицируются только ip-адресами. Карта сети 192.168.1.0 представлена на рис. 2.3.

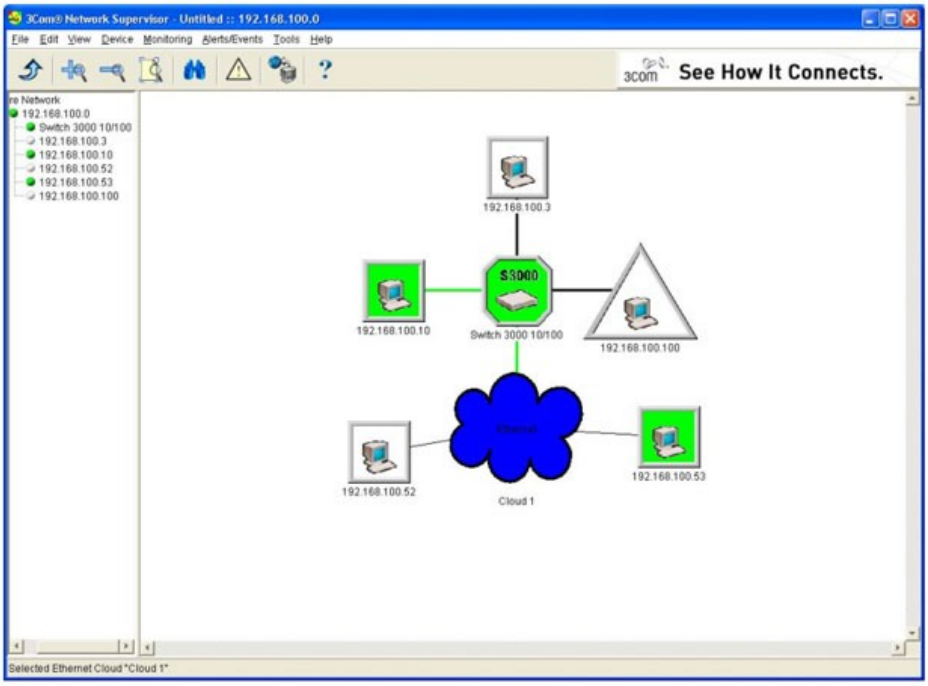


Рис. 2.2. Карта сети 192.168.100.0. Cloud 1 скрывает неуправляемый коммутатор

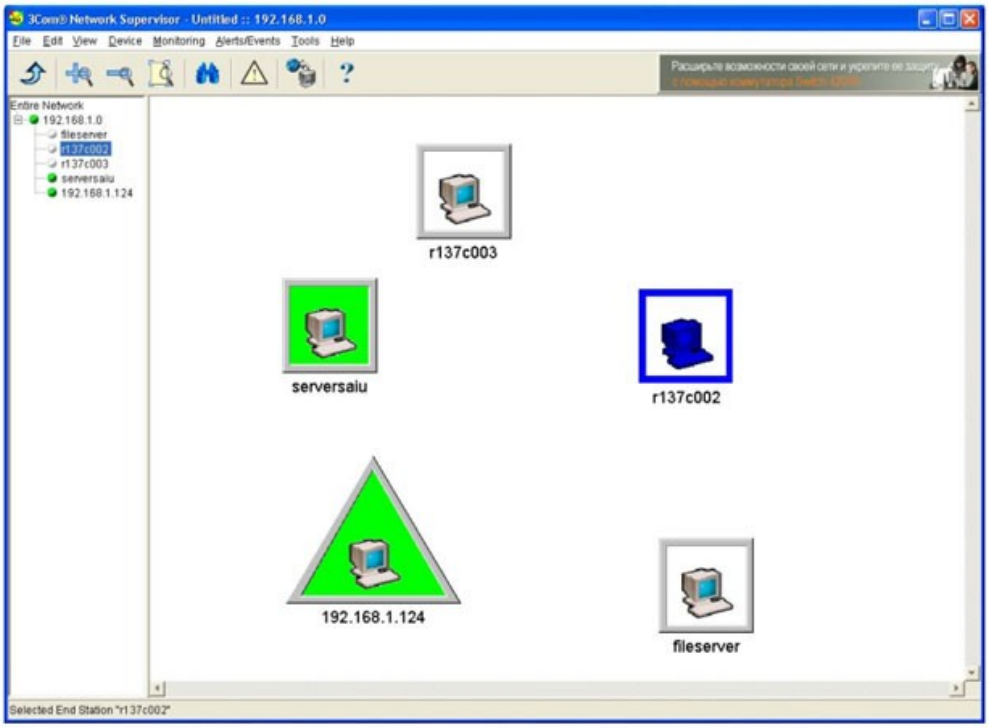


Рис. 2.3. Карта сети 192.168.1.0. Информация от управляемого коммутатора недоступна

Для выбранного узла можно потребовать провести *мониторинг* загрузки различных сетевых сервисов или обратиться к средствам *удаленного администрирования*, использующим протоколы http, telnet или *ssh* (рис. 2.4,2.5).

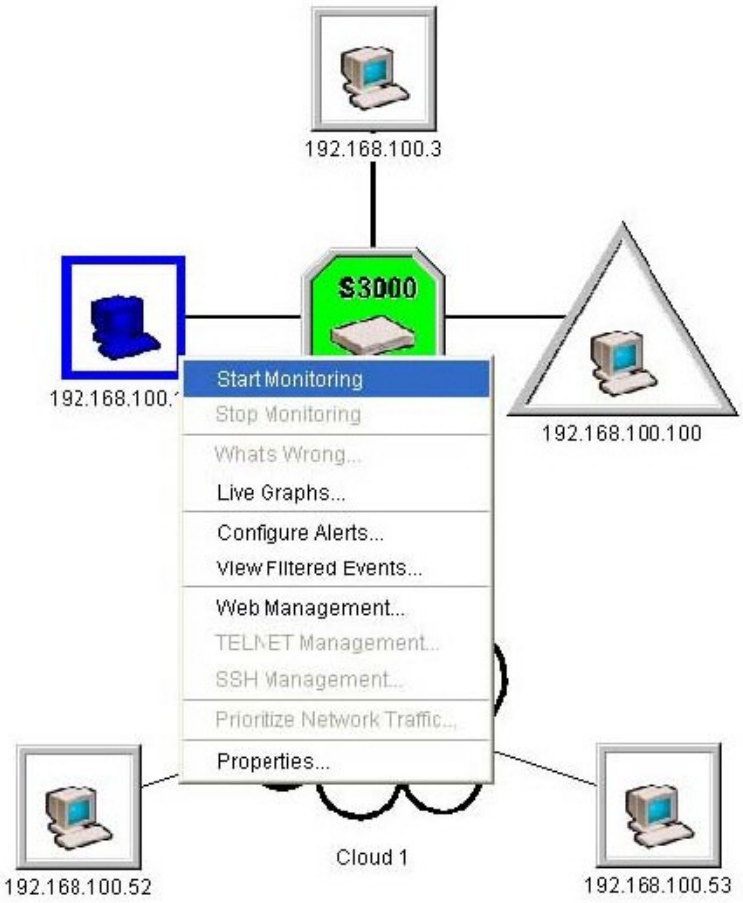


Рис. 2.4. Функции, доступные для выбранного узла

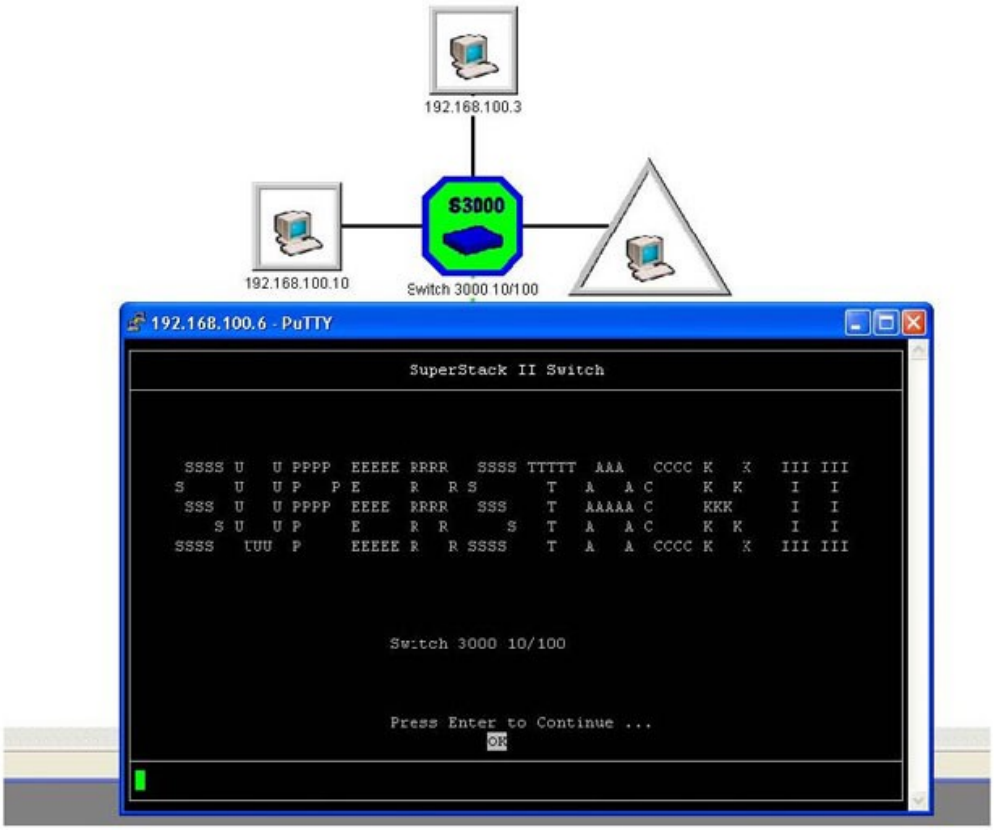


Рис. 2.5. Запуск удаленного терминала для администрирования коммутатора Switch 3000

Функция поиска (кнопка панели инструментов с изображением бинокля) позволяет, в частности, отобразить информацию о типах используемых сетевых подключений (рис. 2.6,2.7). Через свойства управляемого коммутатора доступна *информация* о том, к какому порту какой узел подключен и графики загрузки (рис. 2.8,2.9).

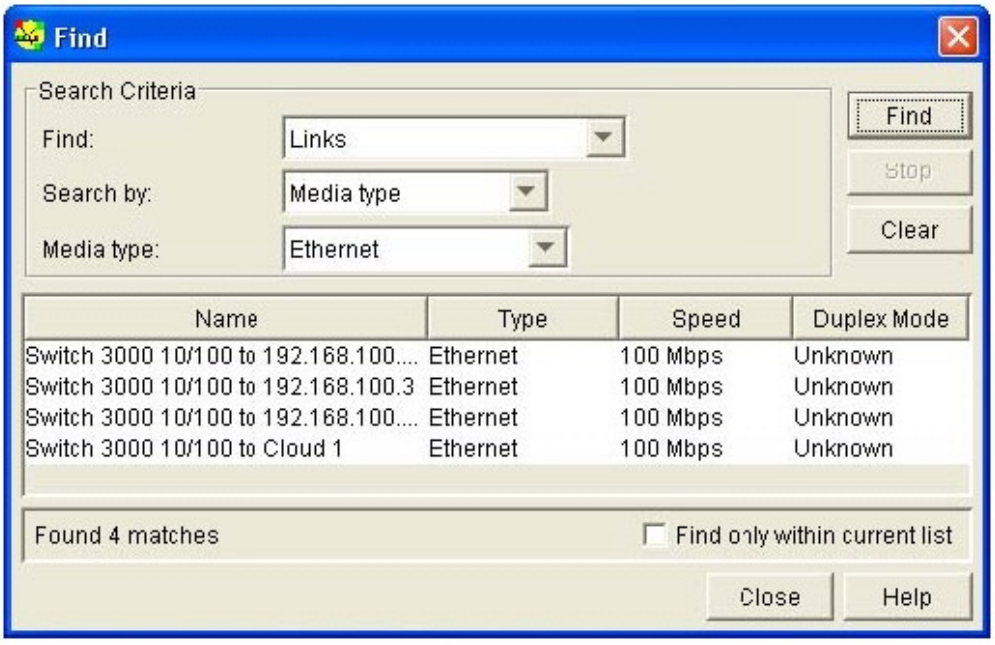


Рис. 2.6. Соединения по типам подключений. Ethernet

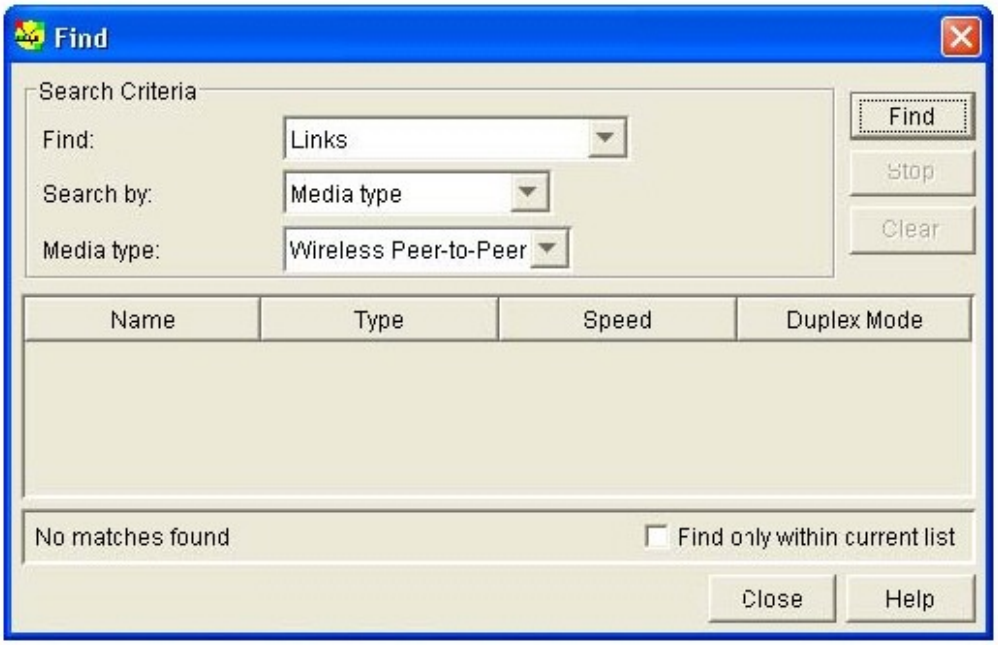


Рис. 2.7. Соединения по типам подключений. Беспроводные подключения (отсутствуют)

Собранная *информация* может отображаться в виде отчетов, формируемых в формате *HTML*. *Опция* доступна через *меню* **Tools пункт Reports**. Для задач, связанных с инвентаризацией системы, наибольший интерес представляют отчеты **Inventory Report** и **Topology Report**.

Отчет *по* топологии сети 192.168.1.0 состоит из записи "Нет данных", т.к. данные о топологии *программа 3Com Network Supervisor* получить не смогла (в этой сети управляемый *коммутатор* "невидим", т.к. его *адрес* принадлежит другой ip-сети).

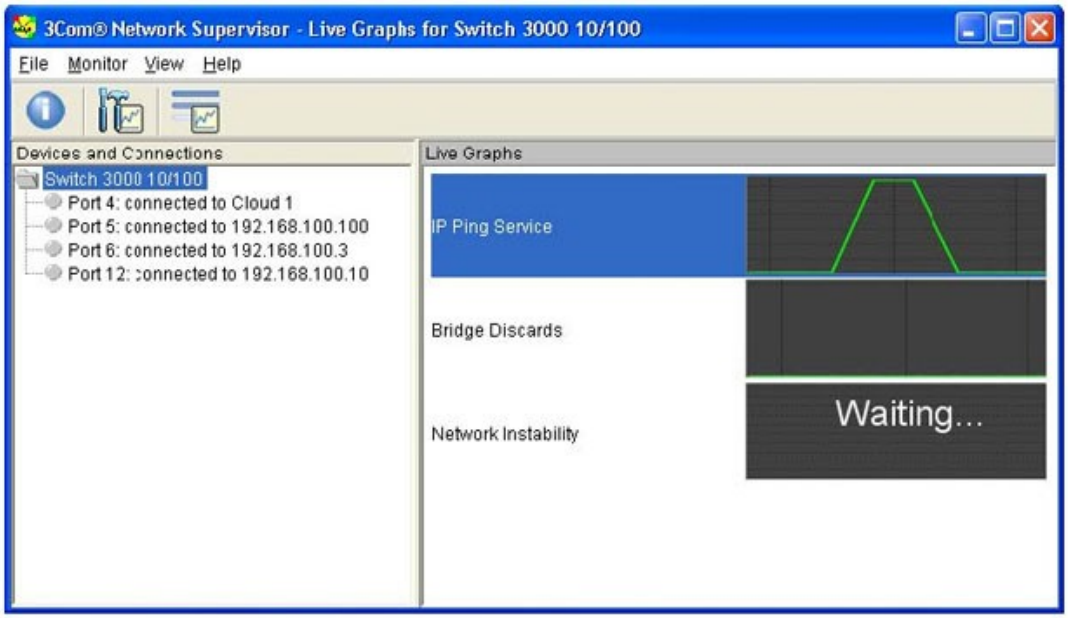


Рис. 2.8. Данные о подключениях и графики

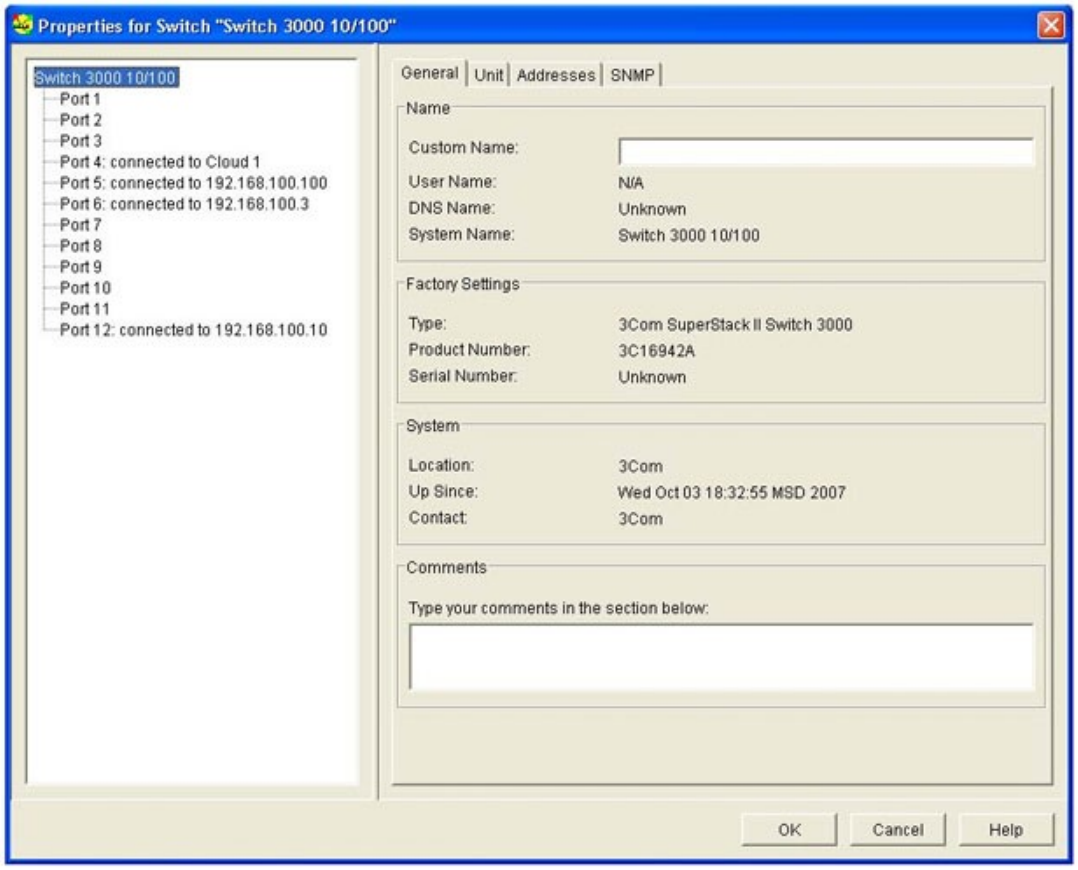


Рис. 2.9. Свойства коммутатора

Задание

С помощью *3Com Network Supervisor* постройте карту сети учебной лаборатории. Опишите узлы сети, используемые типы соединений, доступные средства *удаленного администрирования*.

Перечислите используемые сетевые устройства и укажите, какие последствия будут при выходе из строя (или некорректной работе) каждого из них.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

1.2. ПРАКТИЧЕСКАЯ РАБОТА 3. Идентификация и аутентификация систем семейства Microsoft Windows.

Цель работы:

Изучить способы идентификации и аутентификации систем симейства Microsoft Windows.

Система идентификации и аутентификации. Утилиты Microsoft Baseline Security Analyzer. Двухфакторная аутентификация. Парольные системы аутентификации. Учетная запись пользователя. Задание минимальной длины используемых в системе паролей. Проверка администраторами безопасности качества используемых паролей путем имитации атак. Ограничение числа неудачных попыток ввода пароля. Журнала истории паролей.

Теоретическая часть.

В данном разделе будут рассмотрены некоторые технические меры повышения защищенности систем. Выбор рассматриваемых мер обусловлен возможностью их реализации встроенными средствами операционных систем семейства Microsoft *Windows*. Соответственно, уровень защищенности может быть повышен без дополнительных затрат на специализированные средства защиты.

В теоретической части курса будут методы, лежащие в основе соответствующих средств и механизмов. В лабораторных работах рассматриваются конкретные настройки операционных систем.

Рассматриваемые вопросы можно разделить на две группы:

- вопросы, связанные с *идентификацией и аутентификацией* пользователей;
- защита передаваемых сообщений.

Идентификация и аутентификация

Идентификация - присвоение пользователям идентификаторов (уникальных имен или меток) под которыми система "знает" пользователя. Кроме идентификации пользователей, может проводиться *идентификация* групп пользователей, ресурсов ИС и т.д. *Идентификация* нужна и для других системных задач, например, для ведения журналов событий. В большинстве случаев *идентификация* сопровождается аутентификацией.

Аутентификация - установление подлинности - проверка принадлежности пользователю предъявленного им идентификатора. Например, в начале сеанса работы в ИС *пользователь* вводит имя и *пароль*. На основании этих данных система проводит идентификацию (*по* имени пользователя) и аутентификацию (сопоставляя *имя пользователя* и введенный *пароль*).

Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от несанкционированного доступа (НСД) любой информационной системы. В соответствии с рассмотренной ранее моделью многоуровневой защиты, *аутентификация* пользователя компьютера относится к уровню защиты узлов.

Обычно выделяют 3 группы методов аутентификации.

1. Аутентификация по наличию у пользователя уникального объекта заданного типа. Иногда этот класс методов аутентификации называют по-английски "I have" ("у меня есть"). В качестве примера можно привести аутентификацию с помощью смарт-карт или электронных USB-ключей.

2. Аутентификация, основанная на том, что пользователю известна некоторая конфиденциальная информация - "I know" ("я знаю"). Например, аутентификация по паролю. Более подробно парольные системы рассматриваются далее в этом разделе.

3. Аутентификация пользователя по его собственным уникальным характеристикам - "I am" ("я есть"). Эти методы также называются биометрическими.

Нередко используются комбинированные схемы аутентификации, объединяющие методы разных классов. Например, двухфакторная *аутентификация* - *пользователь* предъявляет системе смарт-карту и вводит пин-код для ее активации.

Наиболее распространенными на данный момент являются парольные *системы аутентификации*. У пользователя есть *идентификатор* и *пароль*, т.е. секретная *информация*, известная только пользователю (и возможно - системе), которая используется для прохождения аутентификации.

В зависимости от реализации системы, *пароль* может быть одноразовым или многоразовым. Операционные системы, как правило, проводят аутентификацию с использованием многоразовых паролей. Совокупность идентификатора, пароля и, возможно, дополнительной информации, служащей для описания пользователя составляют **учетную запись пользователя**.

Если нарушитель узнал *пароль* легального пользователя, то он может, например, войти в систему под его учетной записью и получить *доступ* к конфиденциальным данным. Поэтому безопасности паролей следует уделять особой внимание.

Как отмечалось при рассмотрении стандарта *ISO 17799*, рекомендуется, чтобы пользователи системы подписывали документ о сохранении конфиденциальности пароля. Но нарушитель также может попытаться подобрать *пароль*, угадать его, перехватить и т.д. Рассмотрим некоторые рекомендации *по* администрированию парольной системы, позволяющие снизить *вероятность* реализации подобных угроз.

1. Задание минимальной длины используемых в системе паролей. Это усложняет атаку путем подбора паролей. Как правило, рекомендуют устанавливать минимальную длину в 6-8 символов.

2. Установка требования использовать в пароле разные группы символов - большие и маленькие буквы, цифры, специальные символы. Это также усложняет подбор.

3. Периодическая проверка *администраторами безопасности* качества используемых паролей путем *имитации атак*, таких как подбор паролей "по словарю" (т.е. проверка на использование в качестве пароля слов естественного языка и простых комбинаций символов, таких как "1234").

4. Установка максимального и минимального сроков жизни пароля, использование механизма принудительной смены старых паролей.

5. Ограничение числа неудачных попыток ввода пароля (блокирование учетной записи после заданного числа неудачных попыток войти в систему).

6. Ведение журнала истории паролей, чтобы пользователи, после принудительной смены пароля, не могли вновь выбрать себе старый, возможно скомпрометированный пароль.

Современные операционные системы семейства *Windows* позволяют делать установки, автоматически контролирующие выполнение требований 1,2,4-6. При

использовании домена *Windows*, эти требования можно распространить на все компьютеры, входящие в *домен* и таким образом повысить защищенность всей сети.

Но при внедрении новых механизмов защиты могут появиться и нежелательные последствия. Например, требования "сложности" паролей могут поставить в *тупик* недостаточно подготовленного пользователя. В данном случае потребуется объяснить, что с точки зрения операционной системы *Windows* надежный *пароль* содержит 3 из 4 групп символов (большие буквы, малые буквы, цифры, служебные знаки).

Аналогичным образом, надо подготовить пользователей и к внедрению блокировки учетных записей после нескольких неудачных попыток ввода пароля. Требуется объяснить пользователям, что происходит, а сами правила блокировки должны быть хорошо продуманы. Например, если высока *вероятность* того, что *пользователь* заблокирует свою *запись* в период отсутствия администратора, лучше ставить блокировку не насовсем, а на какой-то срок (30 минут, час и т.д.).

Это приводит к тому, что должна быть разработана политика *управления паролями*, сопровождающие ее документы, а потом уже проведено внедрение.

При использовании ОС семейства *Windows*, выявить учетные записи со слабыми или отсутствующими паролями можно, например, с помощью утилиты Microsoft *Baseline Security Analyzer*. Она же позволяет выявить и другие ошибки администрирования. Вопросам использования этой утилиты, а также настройке политики паролей посвящена лабораторная работа № 3.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.4.ПРАКТИЧЕСКАЯ РАБОТА 4. Аутентификация по протоколу Kerberos.

Цель работы:

Изучение стандарта системы централизованной аутентификации и распределения ключей в рамках операционной системы Windows.

Содержание:

Аутентификация с использованием протокола Kerberos v.5. Централизованное распределение ключей симметричного шифрования. Серверная часть Kerberos как центр распределения ключей. Протокол Kerberos. Взаимодействие между Kerberos-областями.

Теоретическая часть.

Протокол Kerberos был разработан в Массачусетском технологическом институте в середине 1980-х годов и сейчас является фактическим стандартом системы централизованной аутентификации и *распределения ключей* симметричного шифрования. Поддерживается операционными системами семейства Unix, *Windows* (начиная с *Windows*'2000), есть реализации для *Mac* OS.

В сетях *Windows* (начиная с *Windows*'2000 Serv.) *аутентификация по* протоколу Kerberos v.5 (*RFC* 1510) реализована на уровне доменов. Kerberos является основным протоколом аутентификации в домене, но в целях обеспечения совместимости с с предыдущими версиями, также поддерживается протокол *NTLM*.

Перед тем, как рассмотреть порядок работы Kerberos, разберем зачем он изначально разрабатывался. **Централизованное** *распределение ключей* симметричного шифрования подразумевает, что у каждого абонента сети есть только один основной *ключ*, который используется для взаимодействия с *центром распределения ключей* (сервером ключей). Чтобы получить *ключ* шифрования для защиты обмена данными с другим абонентом, *пользователь* обращается к серверу ключей, который назначает этому пользователю и соответствующему абоненту сеансовый *симметричный ключ*.

Протокол Kerberos обеспечивает *распределение ключей* симметричного шифрования и проверку подлинности пользователей, работающих в незащищенной сети. Реализация Kerberos - это программная система, построенная *по* архитектуре "клиент-*сервер*". Клиентская часть устанавливается на все компьютеры защищаемой сети, кроме тех, на которые устанавливаются компоненты сервера Kerberos. В роли клиентов Kerberos могут, в частности, выступать и сетевые серверы (файловые серверы, *серверы печати* и т.д.).

Серверная часть Kerberos называется *центром распределения ключей* (англ. Key *Distribution Center*, сокр. *KDC*) и состоит из двух *компонент*:

- сервер аутентификации (англ. *Authentication Server*, сокр. AS);
- сервер выдачи разрешений (англ. *Ticket* Granting Server, сокр. *TGS*).

Каждому субъекту сети *сервер* Kerberos назначает разделяемый с ним *ключ* симметричного шифрования и поддерживает базу данных субъектов и их секретных ключей. Схема функционирования протокола Kerberos представлена на рис. 4.1.

Пусть клиент C собирается начать взаимодействие с сервером SS (англ. *Service Server* - *сервер*, предоставляющий сетевые сервисы). В несколько упрощенном виде, протокол предполагает следующие шаги:

1. C→AS: {c}.

Клиент C посылает серверу аутентификации AS свой идентификатор c (идентификатор передается открытым текстом).

2. AS→C: $\{\{TGT\}K_{AS\_TGS}, K_{C\_TGS}\}K_C$,

где:

- K_C - основной ключ C ;

- K_{C_TGS} - ключ, выдаваемый C для доступа к серверу выдачи разрешений *TGS* ;
- {TGT} - *Ticket* Granting *Ticket* - билет на доступ к серверу выдачи разрешений

{TGT}={c,*tgs*,t_1,p_1, K_{C_TGS}}, где *tgs* - идентификатор сервера выдачи разрешений, t_1 - отметка времени, p_1 - *период действия* билета.

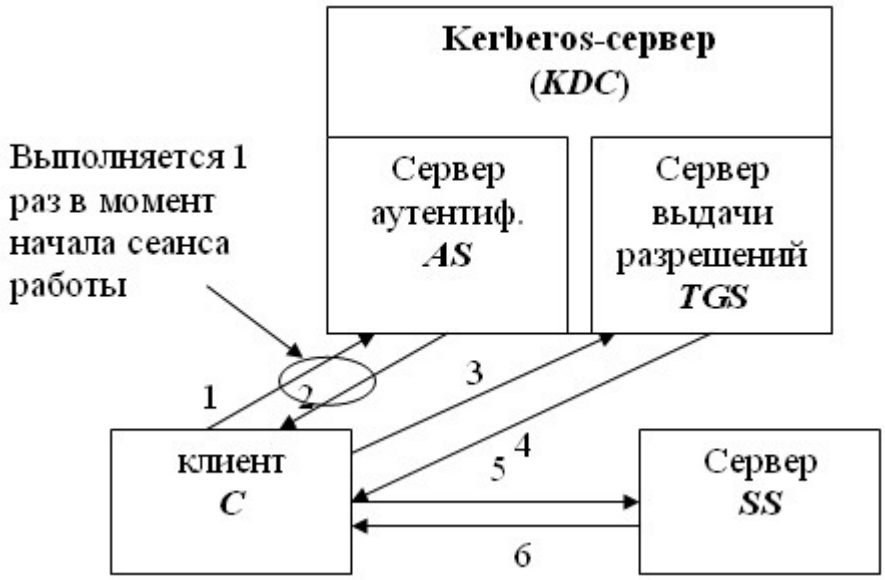


Рис. 4.1. Протокол Kerberos

Запись $\{\cdot\}K_X$ здесь и далее означает, что содержимое фигурных скобок зашифровано на ключе K_X.

На этом шаге сервер аутентификации AS, проверив, что клиент C имеется в его базе, возвращает ему билет для доступа к серверу выдачи разрешений и ключ для взаимодействия с сервером выдачи разрешений. Вся посылка зашифрована на ключе клиента C. Таким образом, даже если на первом шаге взаимодействия идентификатор c послал не клиент C, а нарушитель X, то полученную от AS-посылку X расшифровать не сможет.

Получить доступ к содержимому билета TGT не может не только нарушитель, но и клиент C, т.к. билет зашифрован на ключе, который распределили между собой сервер аутентификации и сервер выдачи разрешений.

3. C→*TGS*: {TGT}K_{AS_TGS}, {Aut_1} K_{C_TGS}, {ID}

где {Aut_1} - аутентификационный блок - Aut_1 = {c,t_2}, t_2 - метка времени; ID - идентификатор запрашиваемого сервиса (в частности, это может быть идентификатор сервера SS).

Клиент C на этот раз обращается к серверу выдачи разрешений TGS. Он пересылает полученный от AS билет, зашифрованный на ключе K_{AS_TGS}, и аутентификационный блок, содержащий идентификатор c и метку времени, показывающую, когда была сформирована посылка. Сервер выдачи разрешений расшифровывает билет TGT и получает из него информацию о том, кому был выдан билет, когда и на какой срок, ключ шифрования, сгенерированный сервером AS для взаимодействия между клиентом C и сервером *TGS*. С помощью этого ключа расшифровывается аутентификационный блок. Если метка в блоке совпадает с меткой в билете, это доказывает, что посылку сгенерировал на самом деле C (ведь только он знал ключ K_{C_TGS} и мог правильно зашифровать свой идентификатор).

Далее делается проверка времени действия билета и времени оправления посылки **3**). Если проверка проходит и действующая в системе политика позволяет клиенту C обращаться к клиенту SS, тогда выполняется шаг **4**).

4. *TGS*→C: $\{\{TGS\}K_{TGS\_SS},K_{C\_SS}\}K_{C\_TGS}$,

где K_{C_SS} - ключ для взаимодействия C и SS, {*TGS*} - *Ticket* Granting Service - билет для доступа к SS (обратите внимание, что такой же аббревиатурой в описании протокола обозначается и сервер выдачи разрешений). $\{TGS\} =\{c,ss,t_3,p_2, K_{C\_SS}\}$.

Сейчас сервер выдачи разрешений *TGS* посылает клиенту C ключ шифрования и билет, необходимые для доступа к серверу SS. Структура билета такая же, как на шаге 2): идентификатор того, кому выдали билет; идентификатор того, для кого выдали билет; отметка времени; *период действия*; ключ шифрования.

5. C→SS: $\{TGS\}K_{TGS\_SS}$, $\{Aut_2\}\ K_{C\_SS}$

где $Aut_2=\{c,t_4\}$.

Клиент C посылает билет, полученный от сервера выдачи разрешений, и свой аутентификационный блок серверу SS, с которым хочет установить сеанс защищенного взаимодействия. Предполагается, что SS уже зарегистрировался в системе и распределил с сервером *TGS* ключ шифрования K_{TGS_SS}. Имея этот ключ, он может расшифровать билет, получить ключ шифрования K_{C_SS} и проверить подлинность *отправителя сообщения*.

6. SS→C: $\{t_4+1\}K_{C\_SS}$

Смысл последнего шага заключается в том, что теперь уже SS должен доказать C свою подлинность. Он может сделать это, показав, что правильно расшифровал предыдущее сообщение. Вот поэтому, SS берет отметку времени из аутентификационного блока C, изменяет ее заранее определенным образом (увеличивает на 1), шифрует на ключе K_{C_SS} и возвращает C.

Если все шаги выполнены правильно и все проверки прошли успешно, то стороны взаимодействия C и SS, во-первых, удостоверились в подлинности друг друга, а во-вторых, получили *ключ* шифрования для защиты сеанса связи - *ключ* K_{C_SS}.

Нужно отметить, что в процессе сеанса работы клиент проходит шаги 1) и 2) только один раз. Когда нужно получить билет на *доступ* к другому серверу (назовем его SS1), клиент C обращается к серверу выдачи разрешений *TGS* с уже имеющимся у него билетом, т.е. протокол выполняется начиная с шага 3).

При использовании протокола Kerberos компьютерная *сеть* логически делится на области действия серверов Kerberos. Kerberos-область - это участок сети, пользователи и серверы которого зарегистрированы в базе данных одного сервера Kerberos (или в одной базе, разделяемой несколькими серверами). Одна область может охватывать сегмент локальной сети, всю локальную *сеть* или объединять несколько связанных локальных сетей. Схема взаимодействия между Kerberos-областями представлена на рис. 4.2.

Для взаимодействия между областями, должна быть осуществлена взаимная *регистрация* серверов Kerberos, в процессе которой *сервер* выдачи разрешений одной области регистрируется в качестве клиента в другой области (т.е. заносится в базу сервера аутентификации и разделяет с ним *ключ*).

После установки взаимных соглашений, клиент из области 1 (пусть это будет K_{11}) может установить *сеанс* взаимодействия с клиентом из области 2 (например, K_{21}). Для этого K_{11} должен получить у своего сервера выдачи разрешений билет на *доступ* к Kerberos-серверу, с клиентом которого он хочет установить взаимодействие (т.е. серверу Kerberos KDC2). Полученный билет содержит отметку о том, в какой области зарегистрирован владелец билета. Билет шифруется на ключе, разделенном между серверами KDC1 и KDC2. При успешной расшифровке билета, удаленный Kerberos-*сервер* может быть уверен, что билет выдан клиенту Kerberos-области, с которой установлены *доверительные отношения*. Далее протокол работает как обычно.

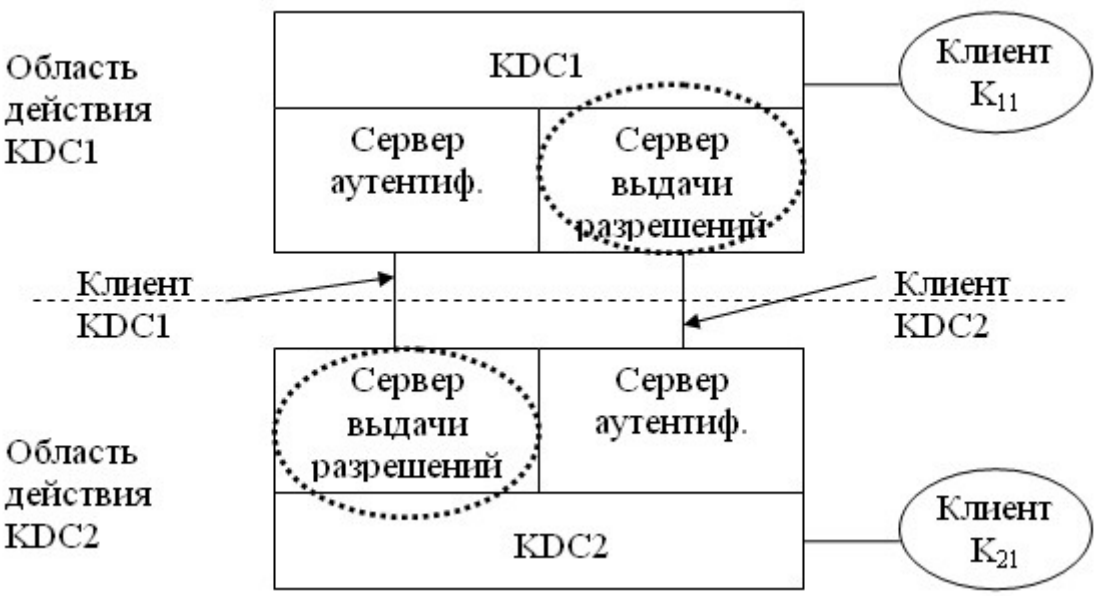


Рис. 4.2. Взаимодействие между Kerberos-областями

Кроме рассмотренных, Kerberos предоставляет еще ряд дополнительных возможностей. Например, указанный в структуре билета *параметр* p(период времени) задается парой значений "время начала действия" - "время окончания действия", что позволяет получать билеты *отложенного действия*.

Имеется тип билета "с правом передачи", что позволяет, например, серверу выполнять действия от имени обратившегося к нему клиента.

Два слова об аутентификации. Если Kerberos - протокол *распределения ключей*, корректно ли использовать его для аутентификации?! Но как отмечалось выше, если все стадии протокола прошли успешно, взаимодействующие стороны не только распределили *ключ*, но и убедились в подлинности друг друга, иными словами - аутентифицировали друг друга.

Что касается реализации протокола Kerberos в *Windows*, то надо отметить следующее.

1. Ключ пользователя генерируется на базе его пароля. Таким образом, при использовании слабых паролей эффект от надежной защиты процесса аутентификации будет сведен к нулю.

2. В роли Kerberos-серверов выступают контроллеры домена, на каждом из которых должна работать служба Kerberos Key Distribution Center (*KDC*). Роль хранилища информации о пользователях и паролях берет на себя служба каталога Active Directory. Ключ, который разделяют между собой сервер аутентификации и сервер выдачи разрешений формируется на основе пароля служебной учетной записи **krbtgt** - эта запись автоматически создается при организации домена и всегда заблокирована.

3. Microsoft в своих ОС использует расширение Kerberos для применения криптографии с открытым ключом. Это позволяет осуществлять регистрацию в домене и с помощью смарт-карт, хранящих ключевую информацию и цифровой *сертификат пользователя*.

4. Использование Kerberos требует синхронизации внутренних часов компьютеров, входящих в домен Windows.

Для увеличения уровня защищенности процесса аутентификации пользователей, рекомендуется отключить использование менее надежного протокола *NTLM* и оставить только Kerberos (если использования *NTLM* не требуют устаревшие клиентские ОС).

Кроме того, рекомендуется запретить администраторским учетным записям получать билеты "с правом передачи" (это убережет от некоторых угроз, связанных автоматическим запуском приложений от имени таких записей).

Контрольные вопросы:

1. Центр распределения ключей (серверная часть Kerberos).
2. Порядок взаимной *регистрации* серверов Kerberos.
3. Основные этапы реализации протокола Kerberos в *Windows*.
4. Служба Kerberos Key Distribution Center.
5. Как осуществляется взаимодействие между Kerberos-областями.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.5.ПРАКТИЧЕСКАЯ РАБОТА 5. Выявление уязвимостей с помощью Microsoft Baseline Security Analyzer.

Цель работы:

Изучить программу Microsoft Baseline Security analyzer, позволяющую проверить уровень безопасности установленной конфигурации операционной системы *Windows*.

Содержание:

Системы анализа защищенности операционных систем. Порядок установки Service pack. Запуск утилиты в режиме проверки обновлений. Уязвимости, связанные с администрированием операционной системы.

Microsoft *Baseline Security analyzer* - *программа*, позволяющая проверить уровень безопасности установленной конфигурации операционной системы (ОС) *Windows 2000*, XP, *Server* 2003, Vista *Server* 2008. Также проверяется и ряд других приложений разработки Microsoft. Данное средство можно отнести к разряду систем *анализа защищенности*. Оно распространяется бесплатно и доступно для скачивания с *web-сервера* Microsoft (*адрес* страницы данной утилиты на момент подготовки описания был: http://technet.microsoft.com/ru-ru/security/cc184924(en-us).aspx).

В процессе работы *BSA* проверяет наличие обновлений безопасности операционной системы, офисного пакета Microsoft Office(для версий XP и более поздних), серверных приложений, таких как MS *SQL Server*, MS *Exchange Server*, *Internet Information Server* и т.д. Кроме того, проверяется ряд настроек, касающихся безопасности, например, действующая политика паролей.

Перейдем к знакомству с программным продуктом. При запуске открывается окно, позволяющее выбрать *объект* проверки - один *компьютер* (выбирается *по* имени или ip-адресу), несколько (задаваемых диапазоном IP-адресов или доменным именем) или просмотреть ранее сделанные отчеты сканирования системы. При выборе сканирования отдельного компьютера *по* умолчанию подставляется имя локальной станции, но можно указать имя или *IP-адрес* другого компьютера.

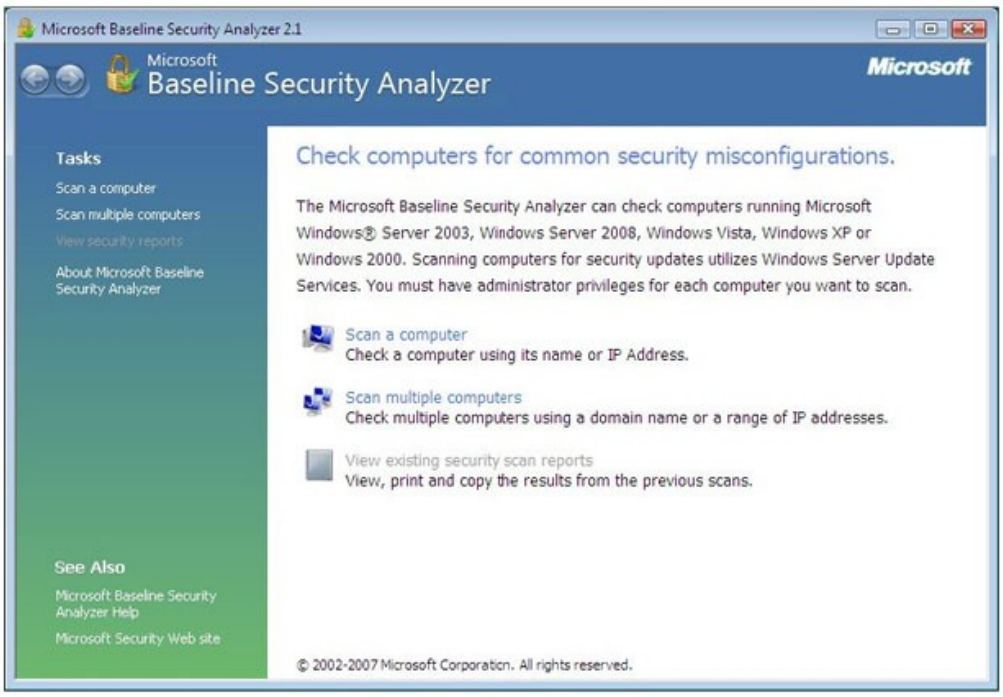


Рис. 5.1. Выбор проверяемого компьютера

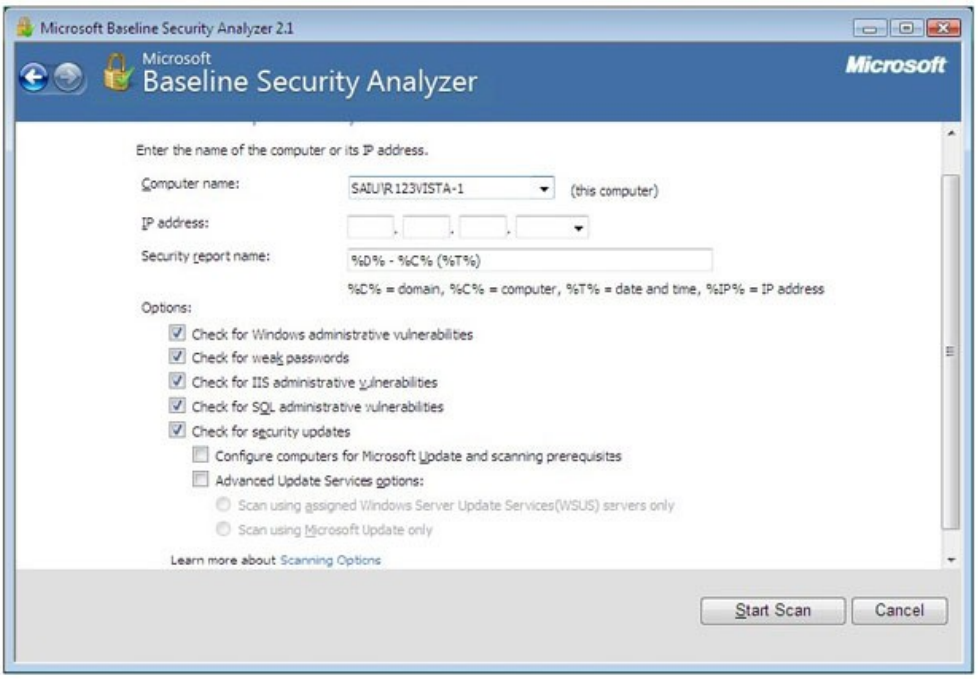


Рис. 5.2. Задание параметров проверки

Можно задать перечень проверяемых параметров. На рис. 5.2 представлен выбор вариантов проверки:

• проверка на наличие уязвимостей Windows, вызванных некорректным администрированием;

• проверка на "слабые" пароли (пустые пароли, отсутствие ограничений на срок действия паролей и т.д.);

• проверка на наличие уязвимостей web-сервера IIS, вызванных некорректным администрированием;

• аналогичная проверка в отношении СУБД MS SQL Server;

• проверка на наличие обновлений безопасности.

Перед началом работы *программа* обращается на *сервер* Microsoft для получения перечня обновлений для ОС и известных уязвимостей. Если на момент проведения проверки *компьютер* не подключен к *Интернет*, база уязвимостей не будет обновлена, *программа* об этом сообщит и дальнейшие проверки выполняться не будут. В подобных случаях нужно отключать проверку обновлении безопасности (сбросив соответствующую галочку на экране рис. 3.2 или с помощью ключа при использовании *утилиты командной строки*, о чем речь пойдет ниже).

Для успешной проверки локальной системы необходимо, чтобы *программа* выполнялась от имени учетной записи с правами локального администратора. Иначе проверка не может быть проведена и о чем будет выдано сообщение: "You do *not* have sufficient *permissions* to*perform this command*. Make sure that you are *running as* the *local administrator* or have opened the *command prompt* using the 'Run *asadministrator*' *option*".

По результатам сканирования формируется отчет, в начале которого дается общая оценка уровня безопасности конфигурации проверяемого компьютера. В приведенном на рис. 5.3 примере уровень риска оценивается как "серьезный" (Severe *risk*).

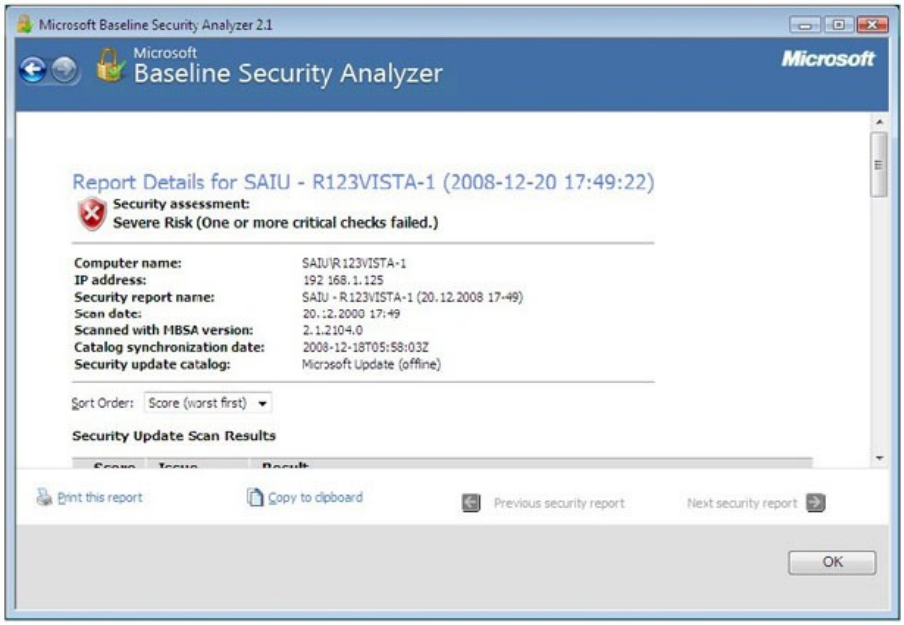


Рис. 5.3. Заголовок отчета

Далее приводится перечень обнаруженных уязвимостей, разбитый на группы: результаты проверки установки обновлений, результаты проверки *Windows* и т.д. Надо отметить, что выпускаемые Microsoft обновления бывают различных типов:

Security updates - собственно обновления безопасности, как правило, посвященные исправлению одной уязвимости программного продукта;

Update rollups - набор исправлений безопасности, который позволяет одновременно исправить несколько уязвимостей. Это упрощает обслуживание процесса обновления программного обеспечения (*ПО*);

Service packs - набор исправлений, как связанных, так и несвязанных с *безопасностью. Установка Service* pack, как правило, исправляет все уязвимости, обнаруженные с момента выхода предыдущего *Service* pack, таким образом устанавливать промежуточные обновления уже не надо.

В описании рассматриваемого результата проверки (рис. 5.4) можно выбрать ссылку **Result details** и получить более подробное описание найденных проблем данной группы. При наличии подключения к *Интернет*, перейдя *по* приводимой в отчете ссылке, можно получить информацию об отсутствующем обновлении безопасности и скачать его из сети.

Нужно отметить, что установка обновлений для систем с высокими требованиями в области непрерывности работы, требует предварительной тщательной проверки совместимости обновлений с использующимися приложениями. Подобная проверка обычно производится на тестовых системах с близкой конфигурацией *ПО*. В то же время, для небольших организаций и пользователей домашних компьютеров такая проверка зачастую неосуществима. Поэтому надо быть готовым к тому, чтобы восстановить систему после неудачного обновления. Для современных ОС семейства *Windows* это можно сделать, например, используя специальные режимы загрузки ОС - безопасный режим или режим загрузки последней удачной конфигурации.

Также надо отметить еще одну особенность. На данный момент **baseline security analyzer** не существует в локализованной русскоязычной версии. И содержащиеся там ссылки на пакеты обновлений могут указывать на иные языковые версии, что может создать проблемы при обновлении локализованных продуктов.

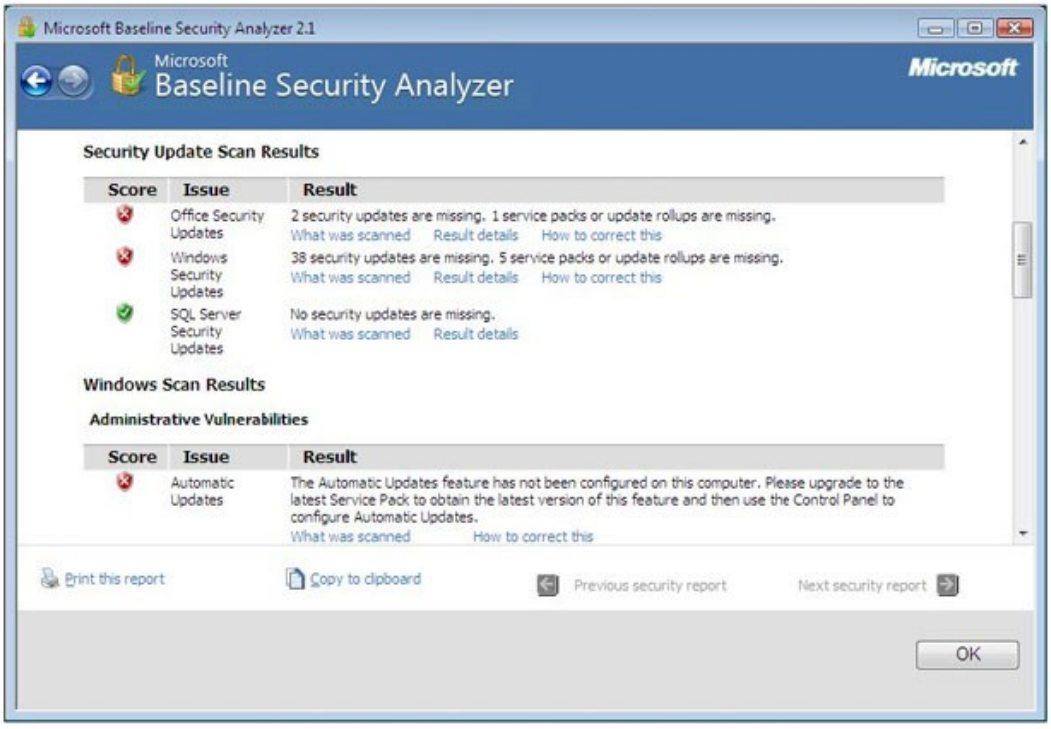


Рис. 5.4. Перечень неустановленных обновлений (по группам)

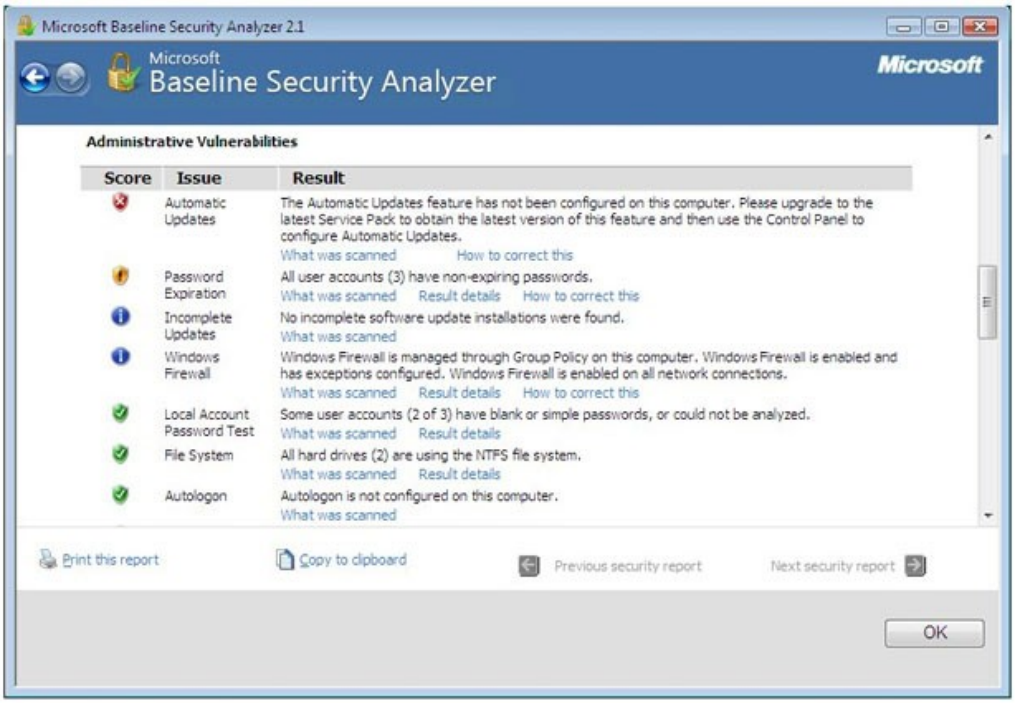


Рис. 5.5. Уязвимости, связанные с администрированием операционной системы

Аналогичным образом проводится работа *по* анализу других групп уязвимостей (рис. 5.5). Описывается *уязвимость*, указывается ее уровень критичности, даются рекомендации *по* исправлению. На рис. 5.6 представлено подробное описание результатов (*ссылка* **result details**) проверки паролей. Указывается, что 3 учетные записи имеют пароли, неограниченные *по* сроку действия.

Рис. 5.6. Результаты проверку паролей

Кроме версии программы с графическим интерфейсом, существует также *утилита* с интерфейсом командной строки. Называется она mbsacli.exe и находится в том же каталоге, куда устанавливался *Baseline security analyzer*, например, **"C:\Program Files\ Microsoft Baseline Security Analyzer 2"**. У утилиты есть достаточно много ключей, получить информацию о которых можно запустив ее с ключом **"/?"**.

Запуск без ключей приведет к сканированию локального компьютера с выводом результатов на *консоль*. Чтобы сохранить результаты сканирования, можно перенаправить *вывод* в какой-либо *файл*. Например: mbsacli > mylog.txt. Хотелось бы еще раз обратить внимание на то, что при настройках *по* умолчанию сначала *утилита* обращается на *сайт* Майкрософт за информацией об обновлениях. Если соединение с *Интернет* отсутствует, то утилиту надо запускать или с ключом /nd (указание "не надо скачивать файлы с сайта Майкрософт") или с ключом/n Updates (указание "не надо проводить проверку обновлений").

Запуск с ключом /xmlout приводит к запуску утилиты в режиме проверки обновлений (т.е. проверка на уязвимости, явившиеся результатом неудачного администрирования, проводиться не будет), при этом, отчет формируется в формате xml. Например:

mbsacli /xmlout > c:\myxmlog.xml

Задания

1. Выполните проверку Вашего компьютера с помощью Microsoft Baseline security analyzer. В отчете о выполнении лабораторной укажите:

   - как оценен уровень уязвимости Вашего компьютера;
   - какие проверки проводились, в какой области обнаружено наибольшее количество уязвимостей;
   - опишите наиболее серьезные уязвимости каждого типа, выявленные на Вашем компьютере.

Проведите анализ результатов - какие уязвимости можно устранить, какие - нельзя из-за особенностей конфигурации ПО или использования компьютера.

2. Выполните удаленную проверку соседнего компьютера из сети лаборатории. Опишите наиболее серьезные уязвимости.

3. Теперь выполните проверку нескольких компьютеров с помощью утилиты mbsacli. Для этого, предварительно создайте текстовый файл с перечнем имен компьютеров или IP-адресов и запускайте mbsacli с ключом /listfile, после которого указывается имя файла с перечнем компьютеров. В результате Вы получите сообщение примерно следующего содержания:

4. Computer Name, IP Address, Assessment, Report Name

5. HOME\MYNBOOK, 127.0.0.1, Severe Risk, HOME - MYNBOOK (06.12.2008 13-51)

Для того, чтобы увидеть подробные результаты проверки, надо повторно запустить mbsacli с ключом /ld, после которого указывается имя отчета. *Вывод* можно перенаправить в *текстовый файл* для дальнейшей обработки. Например:

mbsacli/ld "HOME - MYNBOOK (06.12.2008 13-51)" > c:\test\report1.txt

После выполнения задания проанализируйте результаты, кратко опишите их в отчете *по* практической работе.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.6.ПРАКТИЧЕСКАЯ РАБОТА 6. Настройка локальной политики парольной безопасности операционной системы.

Цель работы:

Изучить методы повышения надежности путем практического применения рекомендаций по администрированию парольной системы операционной системы.

Содержание:

Администрирование парольной системы. Локальная политика безопасности операционной системы. Свойства учетных записей. Политика паролей и учетных записей. Особенности простых и групповых учетных записей.

Теоретический материал.

Локальная политика паролей. Рассмотрим теперь, какие настройки необходимо сделать, чтобы пароли пользователей компьютера были достаточно надежны. В теоретической части курса мы рассматривали рекомендации *по* администрированию парольной системы. Потребовать их выполнения можно с помощью политики безопасности. Настройка делается через **Панель управления Windows**.

Откройте **Панель управления** ⟶ **Администрирование** ⟶ **Локальная политика безопасности**. Выберите в списке **Политика учетных записей** и **Политика паролей**. Для *Windows* Vista экран консоли управления будет выглядеть так, как представлено на рис. 1.

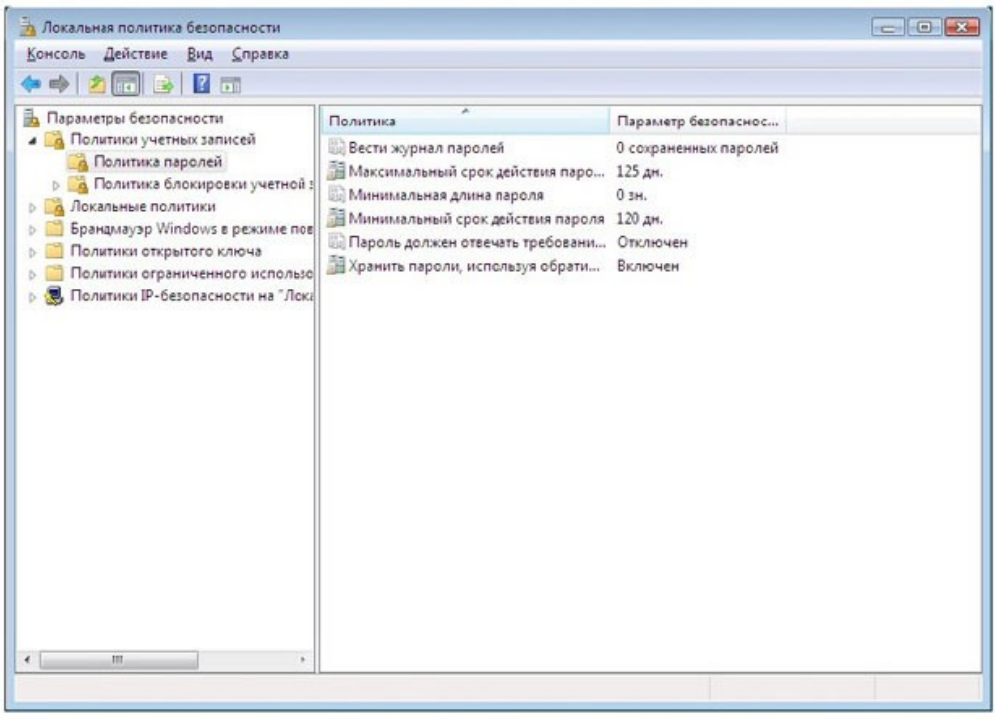


Рис. 1. Настройка политики паролей

Значения выбранного параметра можно изменить (рис. 2). Надо понимать, что не все требования политики паролей автоматически подействуют в отношении всех учетных записей. Например, если в свойствах учетной записи стоит "Срок действия пароля не ограничен", установленное политикой требование максимального срока действия пароля будет игнорироваться. Для обычной пользовательской учетной записи, эту настройку лучше не устанавливать. Но в некоторых случаях она рекомендуется. Например, если в

учебном классе нужна "групповая" учетная *запись*, параметры которой известны всем студентам, лучше поставить для нее **"Срок действия пароля не ограничен"** и **"Запретить смену пароля пользователем"**.

Свойства учетной записи можно посмотреть в **Панель управления** ⟶ **Администрирование** ⟶ **Управление компьютером**, там выберите **Локальные пользователи и группы** и **Пользователи** (или запустив эту же оснастку через **Пуск** ⟶ **Выполнить** ⟶ **lusrmgr.msc**).

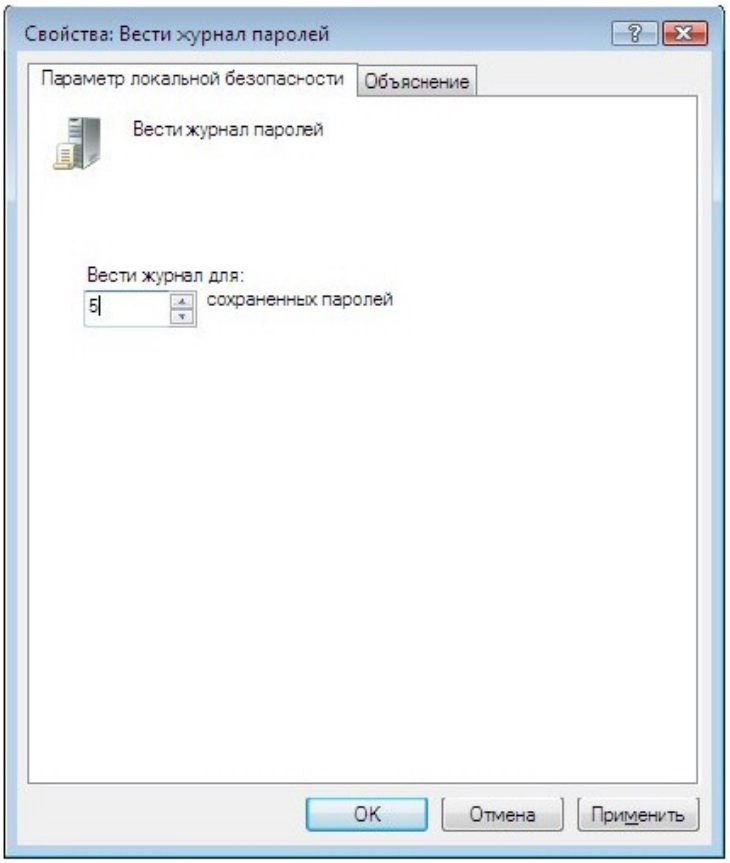


Рис. 2. Установка требования ведения журнала паролей

Задания

1. Опишите действующую на вашем компьютере политику паролей.

2. Измените ее в соответствии с рассмотренными в теоретической части курса рекомендациями по администрированию парольной системы.

Если в ходе проверки утилитой *bsa* были выявлены уязвимости связанные с управлением паролями пользователей, опишите пути их устранения или обоснуйте необходимость использования действующих настроек.

Контрольные вопросы:

1. Что такое политика парольной безопасности?
2. Локальная политика безопасности операционной системы.
3. Администрирование парольной системы..
4. Свойства учетной записи.
5. Особенности простых и групповых учетных записей.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.7.ПРАКТИЧЕСКАЯ РАБОТА 7. Инфраструктура открытых ключей. Цифровые сертификаты.

Цель работы:

Изучить инфраструктуру открытых ключей и цифровые сертификаты.

Содержание:

Инфраструктура открытых ключей (*Public* Key *Infrastructure*, сокр. *PKI*). Центры распределения ключей. Цифровые сертификаты. Проблема аутентификации ключа. Криптографический протокол. Атака типа "человек посередине" (*man* in the middle). Структура PKI. Иерархия центров сертификации и клиентов. Сертификат формата X.509 v.3. Структура списка отозванных сертификатов.

Теоретический материал.

Как было рассмотрено ранее, использование протокола Kerberos позволяет провести аутентификацию и распределить ключи симметричного шифрования. Использование методов асимметричной криптографии сделало возможным безопасный обмен *криптографическими ключами* между отправителем и получателем без использования *центров распределения ключей*.

Но возникает другая проблема - как убедиться в том, что имеющийся у Вас открытый *ключ* другого абонента на самом деле принадлежит ему. Иными словами, возникает проблема аутентификации ключа. Без этого, на криптографический протокол может быть осуществлена *атака* типа "человек посередине" (*man* in the middle).

Идею данной атаки поясняет следующий пример. *Абонент* А (Алиса) хочет послать абоненту В (Боб) зашифрованное сообщение и берет его открытый *ключ* из общедоступного справочника.

Но, на самом деле, ранее нарушитель Е (Ева) подменил в справочнике открытый *ключ* Боба своим открытым ключом. Теперь Ева может расшифровать те сообщения, которые Алиса формирует для Боба, ознакомиться с их содержимым, возможно, зашифровать их на настоящем ключе Боба и переслать ему (рис. 1).

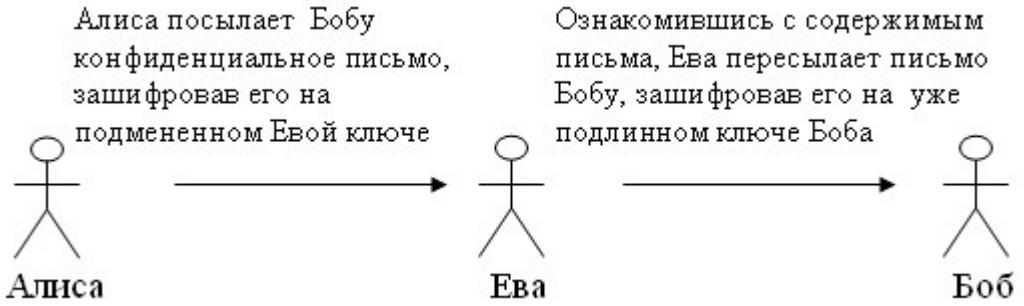


Рис. 1. Атака типа man-in-the-middle

Избежать подобной атаки можно, подтвердив подлинность используемого ключа. Но Алиса и Боб лично никогда не встречались, и передать, например, дискету с ключом из рук в руки не могут. Поэтому, решение задачи подтверждения подлинности берет на себя *третья доверенная сторона* - некий арбитр, которому доверяют оба абонента. Заверяется *ключ* с помощью цифрового сертификата.

На самом деле, подобный способ применяется и вне компьютерных систем. Когда для подтверждения подлинности человека используется паспорт, то в роли третьей

доверенной стороны выступает государство (от имени которого действовали в выдавшем паспорт отделе милиции).

Но вернемся к цифровым сертификатам. Для подтверждения подлинности открытых ключей создается *инфраструктура открытых ключей* (англ. *Public* Key *Infrastructure*, сокр. *PKI*). *PKI* представляет собой набор средств, мер и правил, предназначенных для управления ключами, политикой безопасности и обменом защищенными сообщениями. Структура *PKI* представлена на рис. 2.

Для простоты последующего изложения, будем представлять *сеть* в виде совокупности *удостоверяющих центров* (другое название - *центр сертификации*, от англ. *Certification Authority*, сокр. - *CA*) и пользователей. *Центр сертификации* - *абонент*, которому доверено право удостоверять своей подписью сертификаты, связывающие открытые ключи абонентов с их идентификационной информацией. Сами *центры сертификации* тоже получают сертификаты своих ключей у центров более высокого уровня.

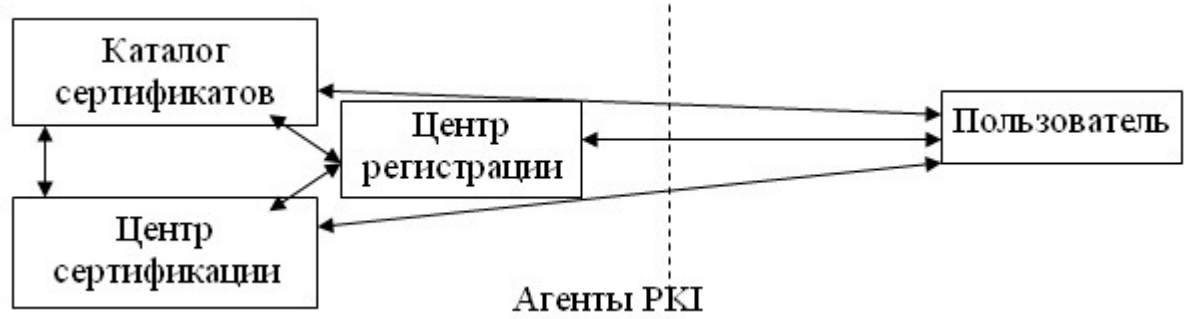


Рис. 2. Структура PKI

Таким образом, *центры сертификации* и пользователи формируют древовидную иерархическую структуру (рис. 3). В вершине этого дерева находится корневой *центр сертификации*, на рисунке - CA_1. Его особенность заключается в том, что он использует *самоподписанный сертификат*, т.е. сам заверяет свой *ключ*.

В приведенном примере, CA_1 заверяет электронной подписью сертификаты подчиненных центров сертификации CA_2 и CA_3, а те, в свою*очередь*, подписывают *сертификаты пользователей* и центров более низкого уровня.

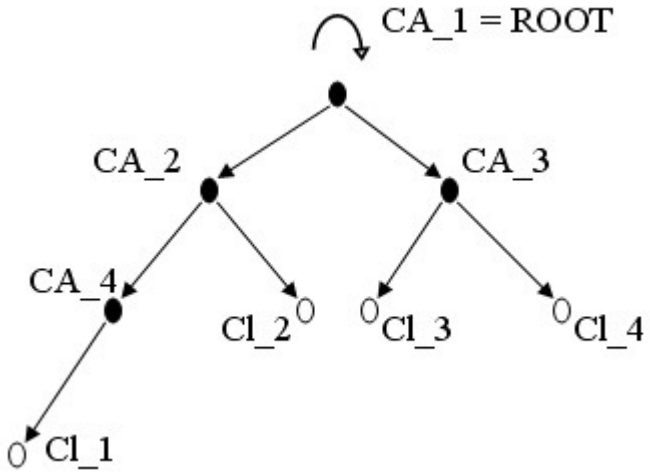


Рис. 3. Иерархия центров сертификации и клиентов

Перейдем к рассмотрению самих сертификатов. Наибольшее распространение получили цифровые сертификаты, формат которых определен стандартом X.509.

На данный момент, принята третья версия стандарта. Формат сертификата изображен на рис. 4.

Номер версии содержит числовое *значение*, соответствующее номеру версии (для сертификата версии 1 равен 0 и т.д.). В первой версии X.509 не было уникальных номеров ("*ID* Изготовителя", "*ID* Субъекта") и *полей расширения*. Во второй версии добавились указанные идентификаторы, в третьей - расширения.

Серийный номер - уникальный номер, присваиваемый каждому сертификату.

Алгоритм подписи - *идентификатор* алгоритма, используемого при подписании сертификата. Должен совпадать с полем **Алгоритм ЭЦП**.

Изготовитель - имя *центра сертификации*, выдавшего сертификат. Записывается в формате *Relative* Distinguished Name - RDN (варианты перевода названия - "относительное отдельное имя", "относительное характерное имя"). Данный формат используется в службах каталога, в частности, в протоколе *LDAP*.

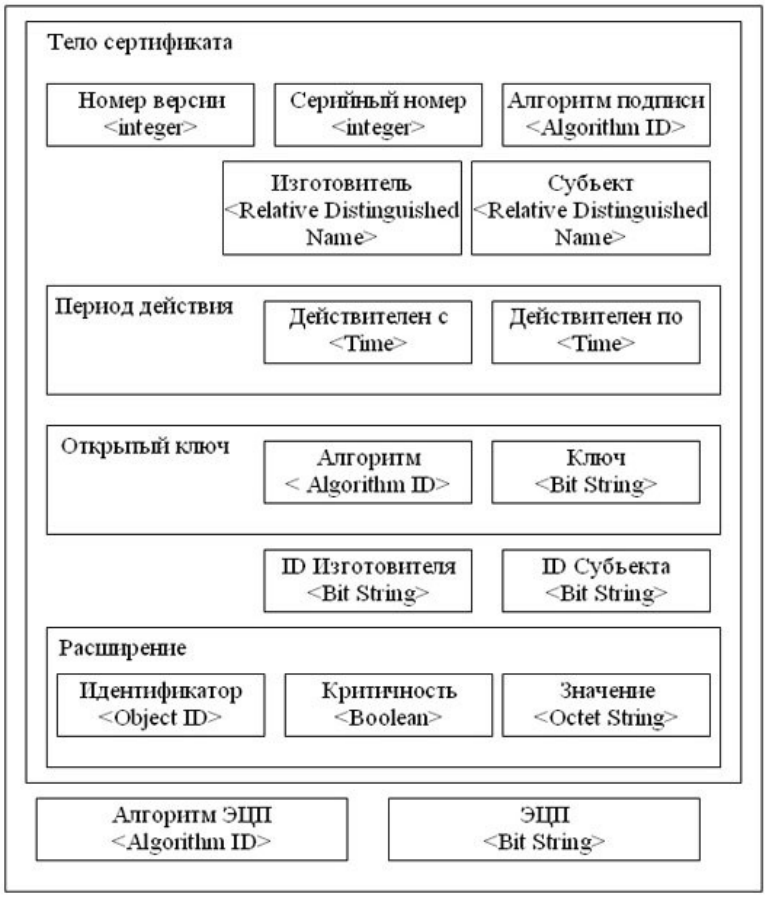


Рис. 4. Сертификат формата X.509 v.3

При записи *Relative* Distinguished Name используются специальные ключевые слова:

- CN (Common Name) - общее имя;
- OU (Organization Unit) - организационная единица;
- DC (Domain Component) - составная часть доменного имени.

Например, в сертификате Microsoft *Windows Hardware Compatibility*, который находится в *хранилище сертификатов Windows*XP *значение* данного поля следующее:

- CN = Microsoft Root Authority
- OU = Microsoft Corporation

- OU = Copyright (c) 1997 Microsoft Corp.

Как видно, указывается имя *центра сертификации*, компания, которой он принадлежит и т.д.

Субъект - имя владельца сертификата, представленное в том же формате RDN. Для указанного в предыдущем примере сертификата значения данного поля:

- CN = Microsoft Windows *Hardware Compatibility*
- OU = Microsoft Corporation
- OU = Microsoft Windows *Hardware Compatibility* Intermediate CA
- OU = Copyright (c) 1997 Microsoft Corp.

Период действия - описывает временной *интервал*, в течение которого *центр сертификации* гарантирует отслеживание *статуса сертификата*(сообщит абонентам сети о факте досрочного отзыва сертификата и т.д.). Период задается датами начала и окончания действия.

Открытый ключ - составное *поле*, содержащее *идентификатор* алгоритма, для которого предназначается данный открытый *ключ*, и собственно сам открытый *ключ* в виде набора битов.

ID Изготовителя и ID Субъекта содержат уникальные идентификаторы *центра сертификации* и пользователя (на случай совпадения имен различных *CA* или пользователей).

Расширения - дополнительный *атрибут*, связанный с субъектом, изготовителем или открытым ключом, и предназначенный для управления процессами сертификации. Более подробно он описан ниже.

Алгоритм электронной цифровой подписи (ЭЦП) - *идентификатор* алгоритма, используемый для подписи сертификата. Должен совпадать со значением поля **Алгоритм подписи**.

ЭЦП - само *значение* электронно-цифровой подписи для данного сертификата.

Расширения могут определять следующие дополнительные параметры:

- идентификатор пары открытый/секретный ключ *центра сертификации* (изготовителя), если центр имеет несколько различных ключей для подписи сертификатов;
- идентификатор конкретного ключа пользователя (субъекта), если пользователь имеет несколько сертификатов;
- назначение ключа, например, ключ для шифрования данных, *проверки ЭЦП* данных, для проверки *ЭЦП* сертификатов и т.д.;
- уточнение периода использования - можно сократить время действия сертификата, указанное в поле *Период действия* (т.е. период, в течение которого статус сертификата отслеживается, станет больше, чем разрешенное время использования сертификата);
- *политики использования* сертификата;
- выбор соответствия *политик использования* сертификата для *центра сертификации* и пользователя, если имеются различные варианты;
- *альтернативное имя* пользователя и *центра сертификации*;

- указания, является ли пользователь сам *центром сертификации* и насколько глубоко разрешается разворачивать *сертификационный путь*.

Предположим, что ключевые пары сгенерированы, открытые ключи заверены сертификатами и размещены в каталоге, реализованном с помощью *web-сервера*, ftp-сервера, службы каталога или другой технологии. Теперь, если *абонент* A желает проверить подпись абонента B под полученным сообщением (или зашифровать для B сообщение с помощью его открытого ключа и т.д.), он выполняет следующие действия:

1. запрашивает в сетевом справочнике сертификат C_B открытого ключа подписи (шифрования,...) абонента B;

2. проверяет достоверность сертификата C_B (см. ниже);

3. в случае успеха проверяет подпись под сообщением (зашифровывает сообщение,...) с помощью открытого ключа, извлеченного из C_B.

Процедура проверки достоверности сертификата C_B состоит в следующем:

1. проверяется срок действия сертификата C_B, если он закончился, сертификат считается недостоверным;

2. из C_B извлекается имя ЦС, подписавшего этот сертификат, обозначим его D ;

3. если D=B, то сертификат самоподписанный, он считается достоверным только, если D=ROOT (хотя, возможно, в некоторых сетях право выдавать самоподписанные сертификаты имеет не один ROOT, это - политика сети);

4. если же $D \neq B$, то из справочника запрашивается сертификат C_D открытого ключа подписи абонента D, проверяется на достоверность сертификат C_D;

5. в случае отрицательного ответа принимается решение о недостоверности сертификата C_B, иначе из C_D извлекается открытый ключ K_D;

6. с помощью K_D проверяется подпись под сертификатом C_B, по результатам проверки этой подписи судят о достоверности C_B.

Если ключи шифрования досрочно вышли из обращения (причины могут быть различны - *пользователь* увольняется из компании, *секретный ключ* скомпрометирован и т.д.), *центр сертификации* извещает об этом остальных пользователей сети путем выпуска списка отозванных сертификатов (англ. *Certificate* Revocation *List*, сокр. *CRL*). Структура *CRL* представлена на рис. 5.

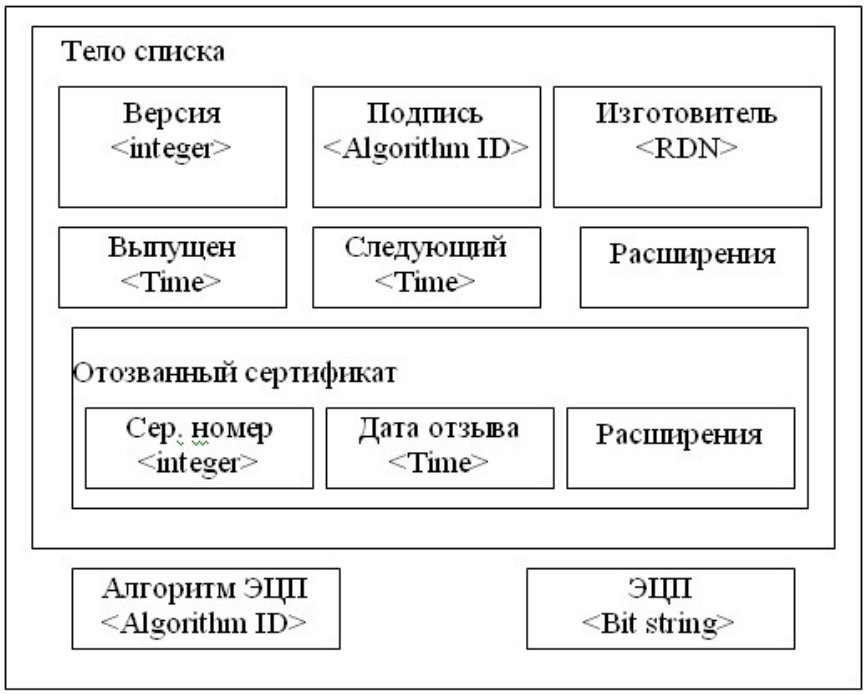


Рис. 5. Структура списка отозванных сертификатов

Поля списка содержат следующую информацию:

- **Номер версии** определяет номер версии формата *CRL*. Текущая используемая версия - вторая.

- **Алгоритм подписи** - идентификатор алгоритма, с помощью которого подписан *CRL*. Должен совпадать по значению с полем **Алгоритм ЭЦП**.

- **Изготовитель** - имя *центра сертификации* в формате RDN.

- **Выпущен** - дата выпуска *CRL*.

- **Следующий** - дата, до которой будет выпущен следующий *CRL*.

- **Расширения** - описывают *центр сертификации*, точку для поиска *CRL* данного центра, номер данного списка и т.д.

- **Отозванный сертификат** - таких полей будет столько, сколько сертификатов отзывается - содержит номер отзываемого сертификата, дату, с которой сертификат отозван, описание причины отзыва.

- **Алгоритм ЭЦП** - идентификатор алгоритма *ЭЦП*, используемого для подписи списка.

- **ЭЦП** - сама *электронная цифровая подпись*.

Проблемы с *CRL* заключаются в том, что может возникнуть ситуация, когда *ключ* уже отозван, но *CRL* еще не выпущен, т.е. пользователи не могут получить информацию о компрометации ключа. Кроме того, распространение *CRL* идет *по* запросу клиента и нарушитель может препятствовать их получению.

Другая проблема *PKI* - *самоподписанные сертификаты*. Сертификат *ROOT* должен раздаваться всем абонентам сети в начале работы и сохраняться в защищенном от подделки хранилище. Иначе нарушитель может попробовать навязать свой сертификат в качестве сертификата корневого центра.

Мы рассмотрели случай реализации **иерархической модели** *PKI*, при которой *центры сертификации* организованы в древовидную структуру с корневым *центром сертификации* на верху иерархии. На практике также встречаются другие варианты организации:

- **одиночный центр сертификации**, который выдает себе *самоподписанный сертификат* - данная модель часто реализуется в небольших организациях, но она имеет отмеченный выше недостаток, связанный с *самоподписанными сертификатами*;

- **одноранговая модель**, при которой независимые *центры сертификации* взаимно сертифицируют друг друга.

Надо отметить, что сфера применения цифровых сертификатов сейчас достаточно широка. В частности, они используются для распределения открытых ключей в протоколах защиты электронной почты *S/MIME* или *PGP*, с помощью цифровых сертификатов проверяется подлинность участников соединения *по* протоколу *SSL* и т.д.

Начиная с *Windows 2000 Server* в состав *серверных ОС* Microsoft входит *программное обеспечение* для создания центров сертификации. Создание корпоративного ЦС может понадобиться, если принято решение использовать защиту электронной почты с помощью *S/MIME*, *шифрование данных* при хранении средствами *EFS* (*EFS* - Encrypted File *System* - реализует *шифрование данных* на дисках с файловой системой *NTFS*), *шифрование* сетевого трафика с помощью протокола IPSec.

Различные практические аспекты использования цифровых сертификатов рассматриваются в лабораторных работах № 6 и 7. Первая из них посвящена работе с сертификатами с точки зрения конечного пользователя (в том числе, получение сертификата для защиты электронной почты с помощью *S/MIME*), а вторая - созданию и администрированию *центра сертификации*. Вопросы использования *EFS* рассматриваются в работе № 8.

Контрольные вопросы:

1. Центры распределения ключей.
2. Инфраструктура открытых ключей.
3. Цифровые сертификаты.
4. Криптографический протокол.
5. Атака типа "человек посередине" (man in the middle).
6. Структура Key Infrastructure.
7. Иерархия центров сертификации и клиентов.
8. Сертификат формата X.509 v.3.
9. Структура списка отозванных сертификатов.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.8.ПРАКТИЧЕСКАЯ РАБОТА 8. Использование цифровых сертификатов.

Цель работы:

Получение и совершенствование практических навыков использования цифровых сертификатов.

Содержание:

Использование протоколов *SSL*/TSL. Центр сертификации VeriSign. Параметры сертификата. Защищенное хранилище ключей и сертификатов в операционной системе Windows. Подключение к системам Интернет-банкинга. Отозванные сертификаты к которым нет доверия. Выбор сертификата для защиты почты с помощью S/MIME в Outlook.

Теоретический материал.

В ходе данной практической работы мы познакомимся с некоторыми вопросами использования цифровых сертификатов.

Начнем с их использования протоколом *SSL*/TSL (на самом деле это два разных протокола, но т.к. TSL разработан на базе *SSL*, принцип использования сертификатов один и тот же). Этот протокол широко применяется в сети *Интернет* для защиты данных передаваемых между web-серверами и браузером клиента. Для аутентификации сервера в нем используется сертификат X.509.

Для примера обратимся на *сайт* Ситибанка (http://www.citibank.ru), в раздел **"Мой банк"**, предназначенный для ведения банковских операций через *Интернет* (рис. 8.1).

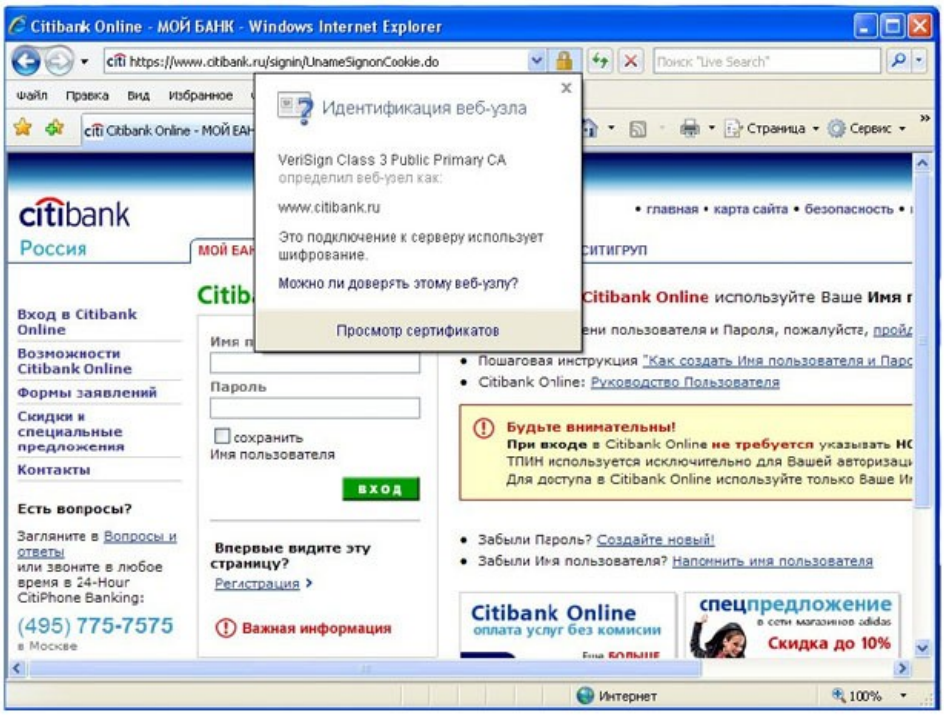


Рис. 8.1. Защищенное соединение

Префикс https в строке адреса и изображение закрытого замка справа от строки указывают, что установлено защищенное соединение. Если щелкнуть мышью *по* изображению замка, то увидим представленное на рис. 8.1 сообщение о том, что подлинность узла с помощью сертификата подтверждает *центр сертификации* VeriSign. Значит, мы на самом деле обратились на *сайт* Ситибанка (а не подделанный нарушителями *сайт*) и можем безопасно вводить логин и *пароль*.

Выбрав **"Просмотр сертификата"** можно узнать подробности о получателе и издателе, другие параметры сертификата (рис. 8.2).

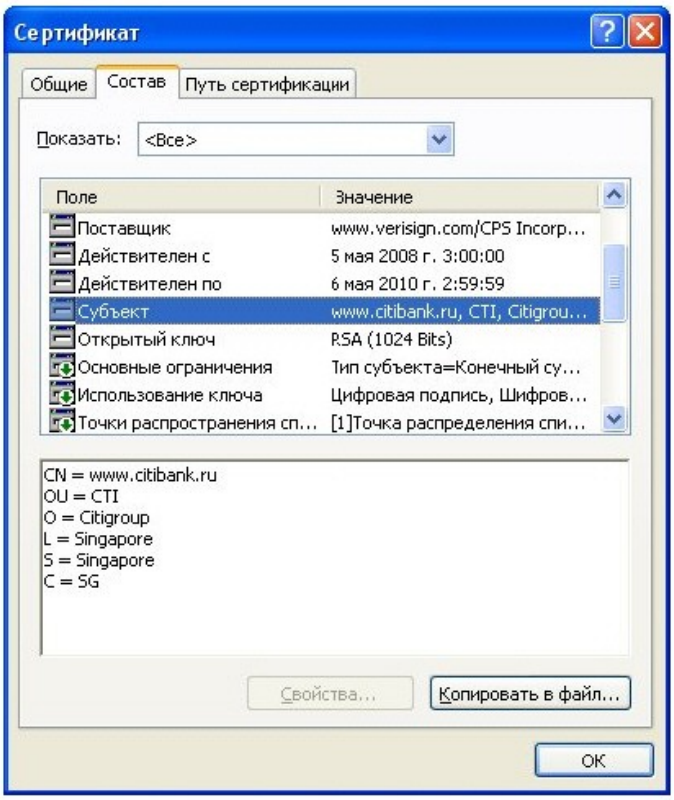


Рис. 8.2. Параметры сертификата

Задание

Посмотрите параметры сертификата "электронной сберкассы" Сбербанка - https://esk.sbrf.ru Опишите, кем на какой срок и для какого *субъекта сертификат* был выдан.

Теперь рассмотрим другой вариант - мы подключаемся *по SSL* к web-серверу, а *браузер* не может проверить его подлинность. Подобная ситуация произошла при подключении в раздел *Интернет*-обслуживания Санкт-Петербургского филиала оператора мобильной связи Tele2 -https://www.selfcare.tele2.ru/work.html (на рис. 8.3).

Если нажать ссылку **"Продолжить открытие этого web-узла"** можно будет просмотреть сертификат.

Задание

Разберитесь, в чем проблема с указанным сертификатом.

Примечание. На всякий случай в конце описания лабораторной приведен ответ.

Теперь рассмотрим, как хранятся сертификаты. *Операционная система Windows* обеспечивает защищенное хранилище ключей и сертификатов. Работать с хранилищем можно используя настройку *консоль* управления *MMC* **"Сертификаты"**.

Из *меню* **Выполнить** запустите *консоль* командой mmc. В *меню* **Консоль** выберите **Добавить или удалить оснастку**, а в списке оснасток выберите **Сертификаты**. Если будет предложен выбор (а это произойдет, если Вы работаете с правами администратора), выберите *пункт* **"Моей учетной записи"**.

Таким образом, мы можем просматривать сертификаты текущего пользователя. Если ранее сертификаты не запрашивались, то в разделе **"Личные сертификаты"** элементов не будет.

В разделе **"Доверенные корневые центры сертификации"** представлен достаточно обширный *список* центров, чьи сертификаты поставляются вместе с операционной системой.

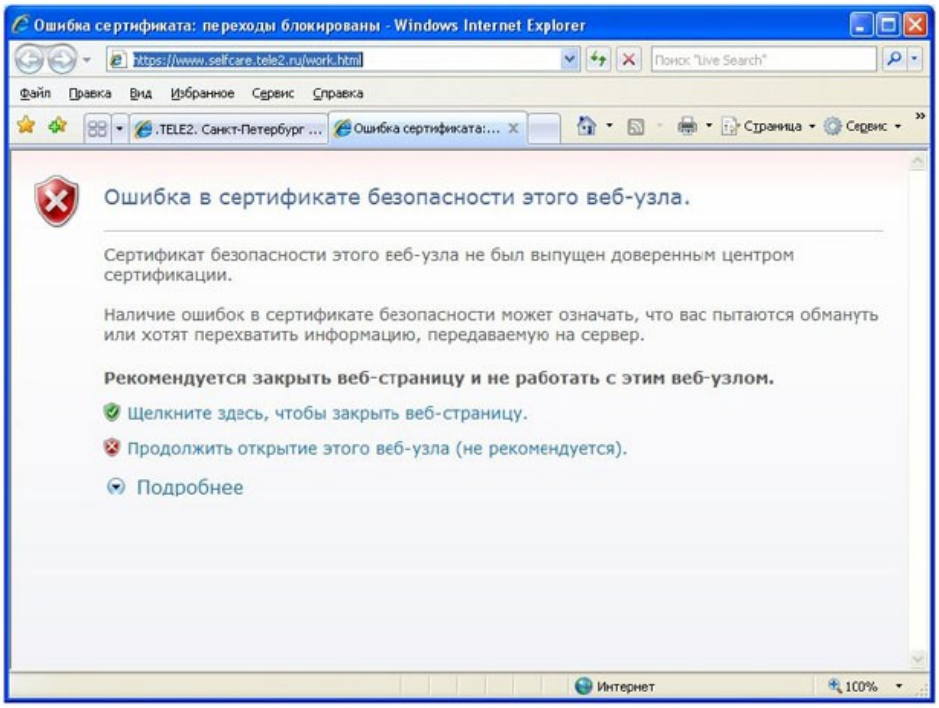


Рис. 6.3. Браузер сообщает о проблеме с сертификатом

Найдите в нем сертификат VeriSign *Class* 3 *Public* Primary *CA*. Благодаря тому, что он уже был установлен, в рассмотренном в начале работы примере с подключением к системам *Интернет*-банкинга *браузер* мог подтвердить подлинность узла.

Теперь перейдем к разделу **"Сертификаты, к которым нет доверия"**. Там находятся отозванные сертификаты. Как *минимум*, там будут находиться два сертификата, которые *по* ошибке или злому умыслу кто-то получил от имени корпорации Microsoft в центре сертификации VeriSing в 2001 году. Когда это выяснилось, сертификаты отозвали (рис. 8.4).

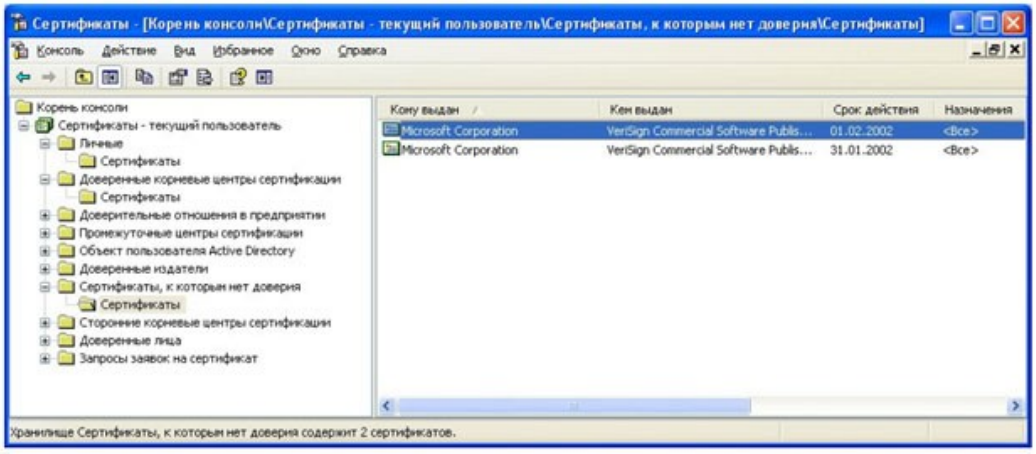


Рис. 8.4. Отозванные сертификаты

Теперь рассмотрим процесс запроса сертификата. На сайте *центра сертификации* Thawte http://www.thawte.com можно бесплатно получить сертификат для электронной почты. Для этого в *меню* сайта **Products** выберите **Free Personal E-Mail Certificates**. После этого надо заполнить небольшую анкету, указав имя, фамилию,

страну, предпочитаемую кодировку, *адрес* электронной почты (должен быть обязательно действующим), дальше - *пароль* и контрольные вопросы для восстановления. Когда все заполнено, на указанный *адрес* почты будет отправлено *письмо* со ссылкой для выполнения дельнейших шагов *генерации ключей* и двумя проверочными значениями, которые нужно ввести, перейдя *по* ссылке. Таким образом, подлинность и принадлежность адреса будет подтверждена.

Далее система предложит ввести *адрес* почты (в качестве *имя пользователя*) и выбранный ранее *пароль*. После чего можно запросить сертификат X.509. Понадобится указать тип браузера и почтового клиента (например, *Internet Explorer* и *Outlook*). После этого потребуется ответить на запросы системы, касающиеся *генерации ключей* (разрешить выполнение ActiveX элемента, выбрать *криптопровайдер*, разрешить генерацию).

После завершения этого этапа на почтовый *адрес* будут выслано второе *письмо*, подтверждающее *запрос* сертификата. А спустя некоторое время - третье, со ссылкой для получения сертификата.

Пройдя *по* ссылке, надо будет снова ввести имя и *пароль* и на странице нажать кнопку **"Install Your Cert"** и согласиться с добавлением сертификата.

В результате в оснастке **Сертификаты** появится личный сертификат выпущенный издателем Thawte *Personal* Freemail *Issuing CA* для субъекта Thawte Freemail *Member* с указанным вами адресом почты (рис. 8.5).

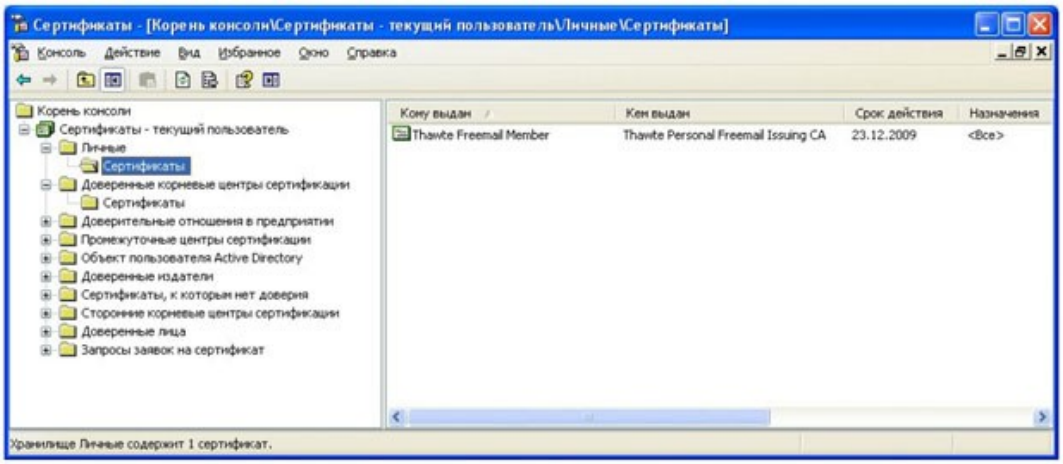


Рис. 8.5. Полученный сертификат

Если использовать сертификат для защиты почты, дальнейшая настройка зависит от почтового клиента. Если это Microsoft *Outlook*, можно использовать встроенную в него поддержку протокола *S/MIME*. В *Outlook* 2003 для выбора сертификата надо войти в *меню* **Сервис Параметры**, там выбрать вкладку **Безопасность** и там в параметрах шифрованной электронной почты выбрать используемый сертификат и алгоритмы (рис. 8.6).

Задание

Запросите сертификат в Thawte и настройте *почтовый клиент* для использования *S/MIME*.

Проблема была в том, что сертификат "самоподписанный": он был выдан *центром сертификации* http://www.selfcare.tele2.ru самому себе. *Браузер* сообщает о невозможности удостовериться в подлинности узла из-за того, что данный *центр сертификации* отсутствует в списке доверенных, а проверить его подлинность с помощью "вышестоящего" по иерархии центра не представляется возможным (т.к. вышестоящего центра нет). Доверять или нет такому сертификату – каждый решает самостоятельно.

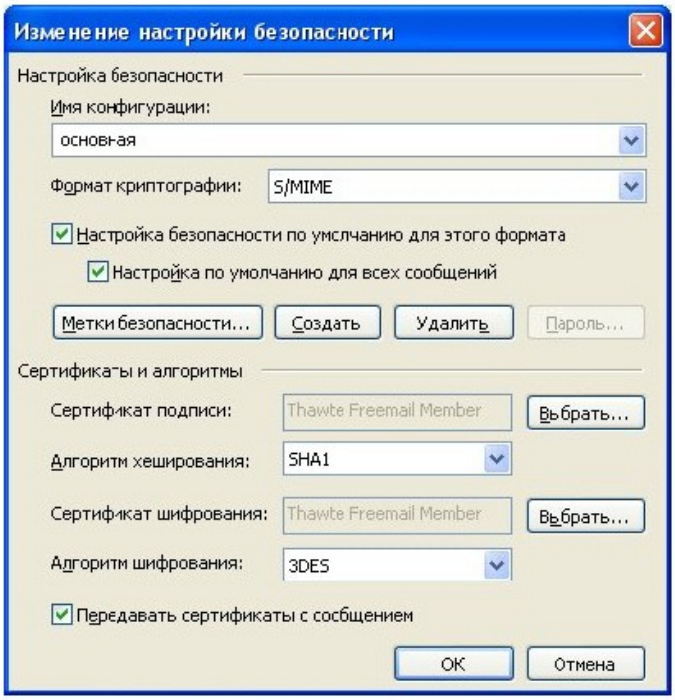


Рис. 8.6. Выбор сертификата для защиты почты с помощью S/MIME в Outlook

Контрольные вопросы:

1. Использование протоколов *SSL*/TSL.
2. Центр сертификации VeriSign.
3. Защищенное хранилище ключей и сертификатов в операционной системе Windows.
4. Подключение к системам Интернет-банкинга.
5. Отозванные сертификаты к которым нет доверия.
6. Выбор сертификата для защиты почты с помощью S/MIME в Outlook.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

3.9.ПРАКТИЧЕСКАЯ РАБОТА 9. Резервное копирование в Windows Server 2008.

Цель работы:

Познакомиться со средствами организации резервного копирования в операционной системе Microsoft *Windows Server* 2008.

Содержание:

Утилиты администрирования. Полное резервное копирование и копирование отдельных дисков. Выбор дисков для резервного копирования. Доступные резервные копии для выбранного сервера. Выбор типа и параметров восстановления. Диск для хранения резервных копий. Выбор типа резервного копирования для диска.

Теоретическая часть

С точки зрения управления рисками, важность процедуры резервного копирования очень высока. В тех случаях, когда реализация угрозы приводит к изменению или удалению данных, повреждению программных *компонент* системы, *резервное копирование* позволяет снизить причиненный *ущерб* и значительно ускорить восстановление системы. При разработке политики резервного копирования нужно определить, как *минимум*, следующие параметры:

- частоту выполнения резервных копий;
- порядок восстановления данных из резервных копий;
- объем носителей информации, выделяемых для хранения резервных копий;
- количество хранимых копий;
- вопросы обеспечения безопасности носителей резервных копий.

Утилиты резервного копирования *Windows Server* 2008 существенно отличаются от того, что было в *Windows Server* 2003 (где эти задачи решались с помощью утилиты ntbackup). Чтобы их использовать, для начала требуется их установить (*по* умолчанию, они не устанавливаются). Делается это с помощью оснастки **Server Manager**, где надо выбрать *пункт* **Add Feature** в разделе **Features** (рис. 9.1) и в появившемся списке выбрать *пункт* **Windows Server Backup Features** (рис. 9.2).

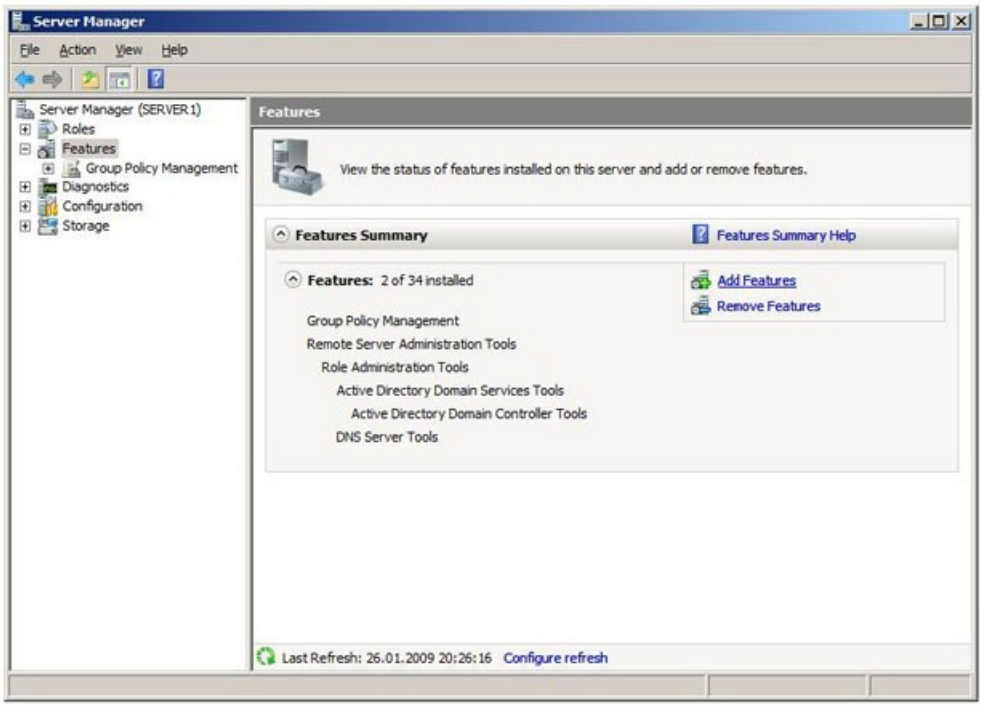


Рис. 9.1. Оснастка Server Manager позволяет добавить компоненты

Как видно на рис. 9.2, предлагается выбрать следующие опции:

- Windows Server Backup;
- Command-line tools (*утилиты командной строки*).

Установка последних, позволяет управлять резервным копированием с помощью сценариев и требует установки *Windows* PowerShell. Но для выполнения лабораторной будет достаточно установить только *Windows Server Backup*.

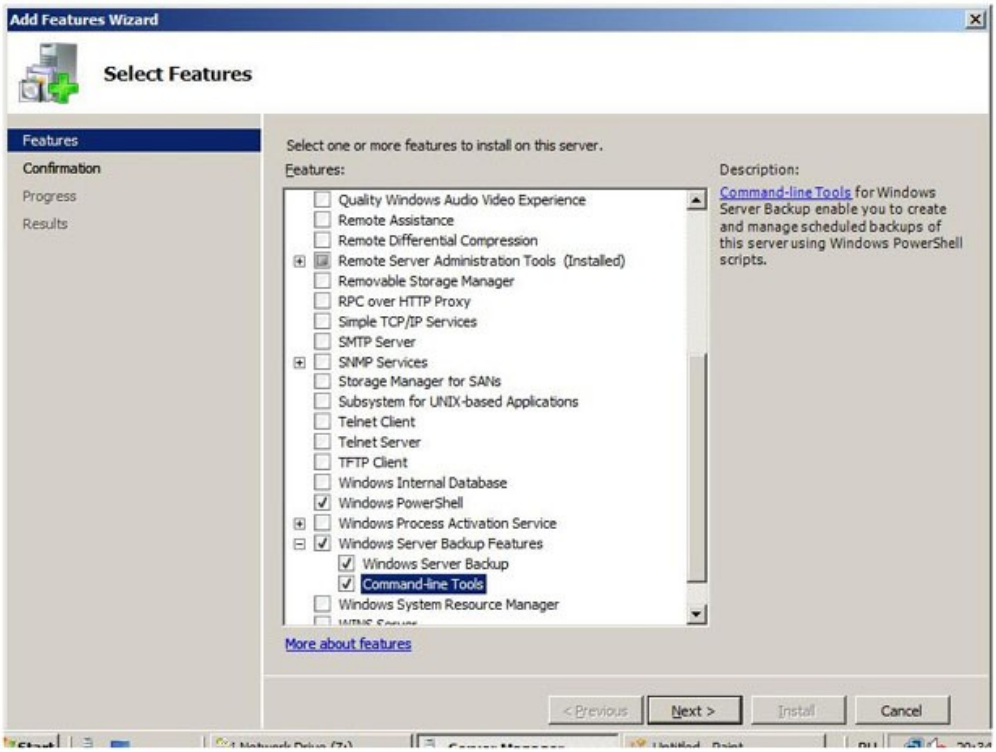


Рис. 9.2. Добавляем утилиты администрирования

После установки, в *меню* **Administrative Tools** становится доступной оснастка **Windows Server Backup**. С ее помощью можно проводить *резервное копирование* данных на локальном или удаленном компьютере (если это разрешено настройками).

Рассмотрим, как это происходит. Запустим утилиту. *Резервное копирование* может проводить *пользователь*, состоящий в группе **Administrators** (Администраторы) или **Backup Operators** (*Операторы* архива). При этом, у членов группы **Backup Operators** при запуске оснастки **Windows Server Backup** будет дополнительно запрашиваться *пароль* (в окне **User Account Control**), т.к. эти *операции* относятся к разряду потенциально опасных.

В окне оснастки в списке доступных действий (**Actions**), расположенном в правой части экрана, выберем опцию **Backup Once …** (т.е. однократная *архивация*). Запустившийся мастер резервного копирования предложит выбор между настройками для уже запланированного копирования (**The same options that you used in the Backup Schedule Wizard for scheduled backups**) и новыми (**Different options**). Нужно выбрать второй вариант (если, как в нашем примере, *утилита* ранее не использовалась, то первый *пункт* списка будет неактивен).

Следующее окно мастера позволяет выбрать, производить ли полное *резервное копирование* или *копирование* отдельных разделов (рис. 9.3). Здесь проявляется первое отличие новых инструментов - *резервное копирование* отдельных папок и файлов производить нельзя, только *логический диск* целиком.

Хотелось бы также обратить внимание на надпись в нижней части экрана, там дается *ссылка* на раздел справки, описывающий выполнение с помощью *утилиты командной строки* резервного копирования только состояния системы (**System State**).

Выберем вариант **Custom**.

Тогда на следующем экране появится *список* дисков (рис. 9.4). Устанавливая или снимая отметки, можно указать, данные с каких дисков помещаются в резервную копию. *Опция* **Enable System Recovery** включает в *архив разделы*, где находятся компоненты операционной системы и файлы необходимые для загрузки (т.е. отметку напротив этих разделов будет не снять).

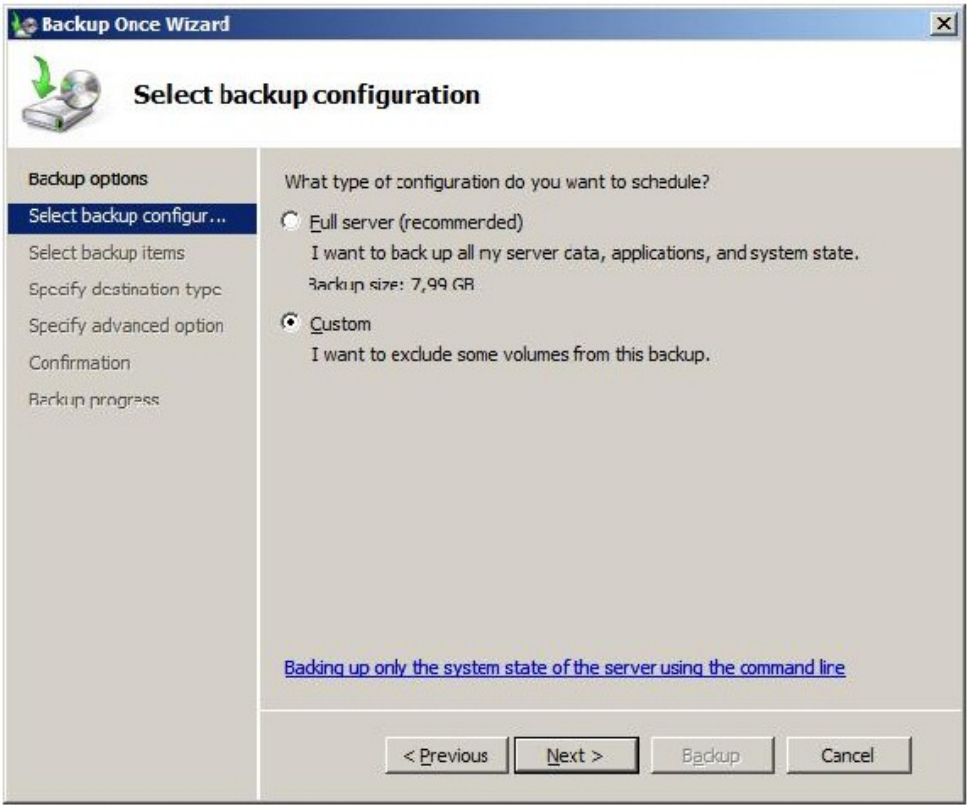


Рис. 9.3. Выбор между полным резервным копированием и копированием отдельных дисков

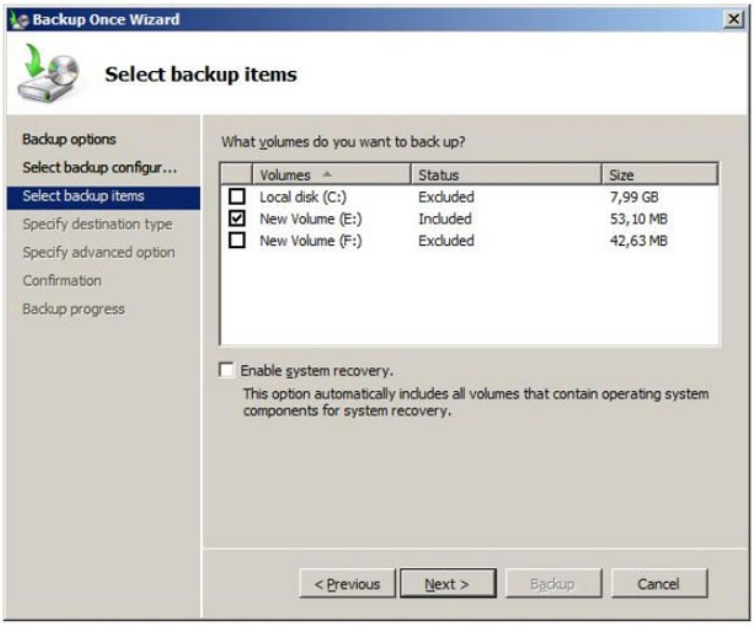


Рис. 10.4. Выбор дисков для резервного копирования

Предположим, нам нужно сделать резервную копию диска E:, на котором находятся пользовательские данные. Тогда отметки устанавливаем так, как это сделано на рис. 9.4 и переходим к следующей стадии, на которой нужно определить, куда будет производиться *копирование*. Это может быть локальный *диск* (жесткий *диск*, пишущий DVD-привод и т.д.) или сетевая *папка*. Надо учитывать, что архивная копия не может

сохраняться на *диск*, входящий в перечень архивируемых. Также нельзя сохранить *архив* на *диск*, где хранятся файлы операционной системы.

Учитывая все вышеизложенное, в рассматриваемом примере можно сделать резервную копию диска E: на *диск* F:, в сетевую папку или на DVD-*диск*. Выберем первый вариант, что и укажем в следующем окне мастера. После чего будет предложено выбрать тип резервного копирования (рис. 9.5).

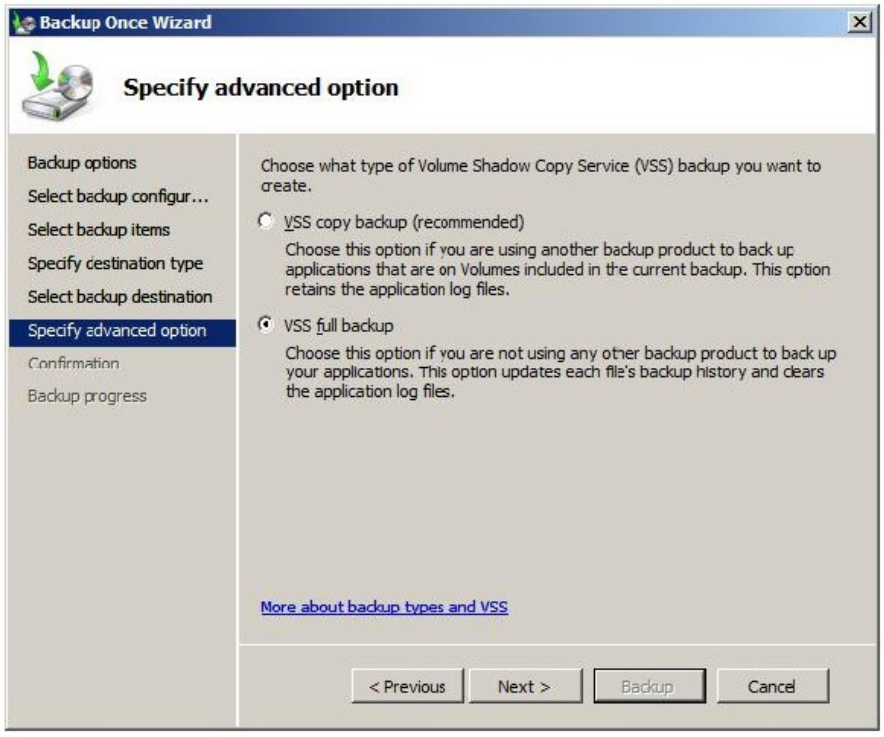


Рис. 9.5. Выбор типа копирования

Служба *Volume* Shadow *Copy Service* (*VSS*) может при резервном копировании отмечать файлы, как помещенные в *архив*, или не делать это. Если кроме средств *Windows Server* 2008 используются и другие продукты для резервного копирования, рекомендуется выбрать вариант **VSS copy backup**. Если такого нет, можно смело выбирать вариант **VSS full backup**.

В следующем окне мастера будет запрошено подтверждение и, если оно получено, запустится *резервное копирование*.

В результате, в нашем примере на диске F: появится каталог **WindowsImageBackup**, в нем будет создан *подкаталог*, названный *по* имени архивируемого сервера, куда и попадет копия.

Задание

1. На учебном сервере (или виртуальной машине) выберите раздел для резервного копирования.

2. С учетом рассмотренных ограничений и объема копируемого раздела, выберите место для размещения копии. Определите, от имени какой учетной записи будет проводиться эта операция.

3. Выполните однократное резервное копирование выбранного раздела.

Теперь рассмотрим порядок восстановления данных из резервной копии.

В первой части лабораторной работы была сделана резервная копия раздела E:. Пусть понадобилось восстановить содержимое одной из папок из этого раздела. При этом требуется сравнить текущее содержимое папки с архивной копией, т.е. восстанавливать нужно в другую папку.

Запускаем оснастку **Windows Server Backup** и в списке **Actions** выбираем **Recover** (восстановление). Мастер восстановления уточняет, какой *сервер* будет восстанавливаться, после чего представит перечень имеющихся резервных копий (рис. 9.6).

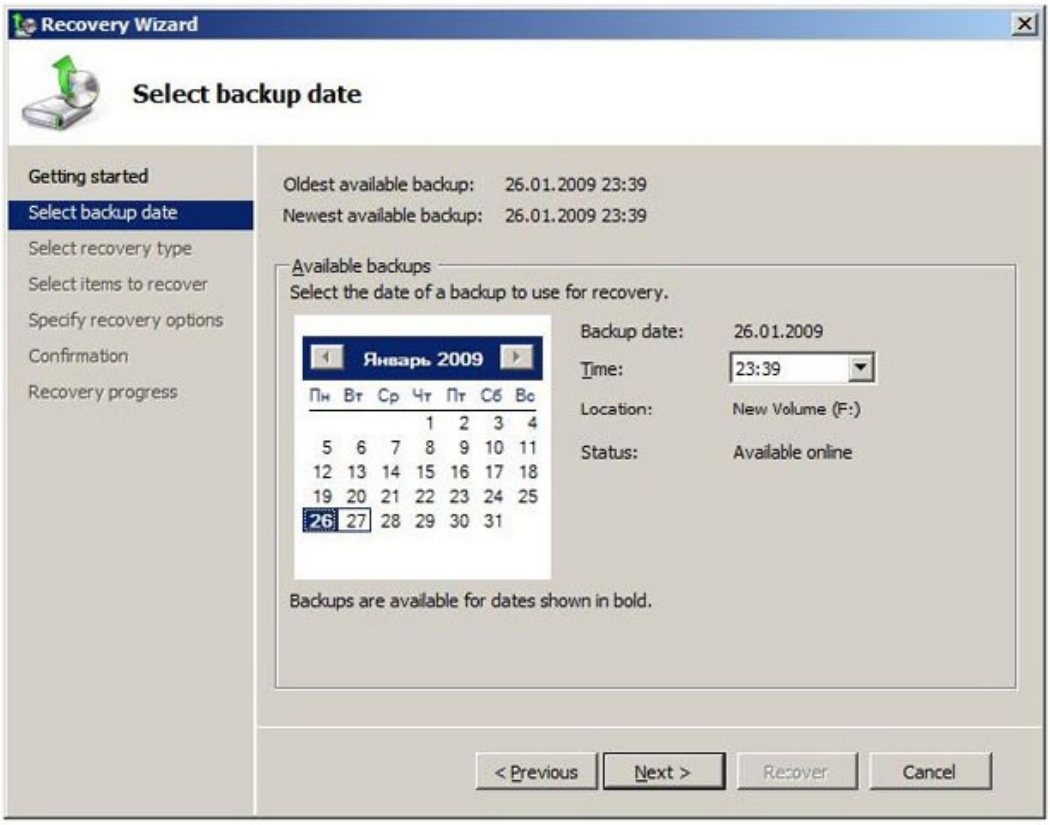


Рис. 9.6. Перечень доступных резервных копий для выбранного сервера

В следующем окне запрашивается, что именно восстанавливается. Нас интересует отдельная *папка*, потому выбираем вариант **Files and folders** (рис. 9.7). Другие варианты - восстановление зарегистрированных приложений и восстановление *раздела диска* целиком.

В следующем окне мастера в выпадающем списке нужно найти и выделить выбранную для восстановления папку. Если восстановить нужно несколько объектов, их выделяют совместно, удерживая клавишу **Ctrl** (или **Shift** для выделения диапазона). После этого выбирается *путь* для восстановления и задаются параметры. В нашем примере, мы хотим восстановить выбранную папку с файлами во вновь созданную папку **restored** (рис. 9.8).

Кроме пути (исходный или альтернативный), выбирается вариант действий при совпадении имен файлов и папок. Это особенно актуально, если восстанавливать файлы в исходную папку. Вариантов три - создавать копии, перезаписывать имеющиеся объекты восстанавливаемыми, оставить имеющиеся объекты.

Последний из выбираемых в этом окне параметров указывает на то, восстанавливать ли настройки безопасности (т.е. списки доступа к файлам).

После выбора всех параметров будет запрошено подтверждение и начнется восстановление.

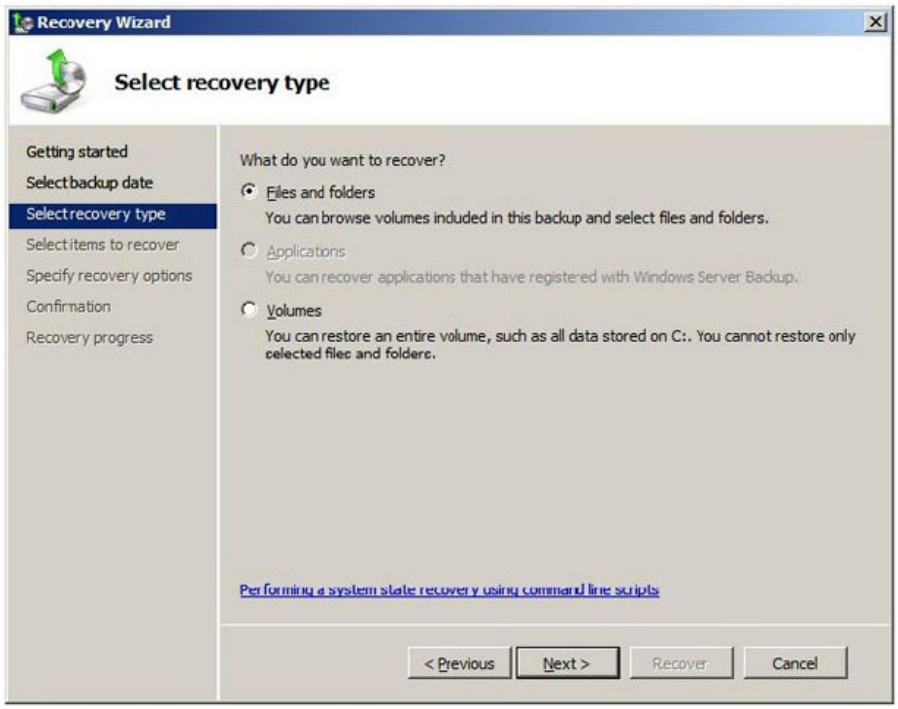


Рис. 9.7. Выбор типа восстановления

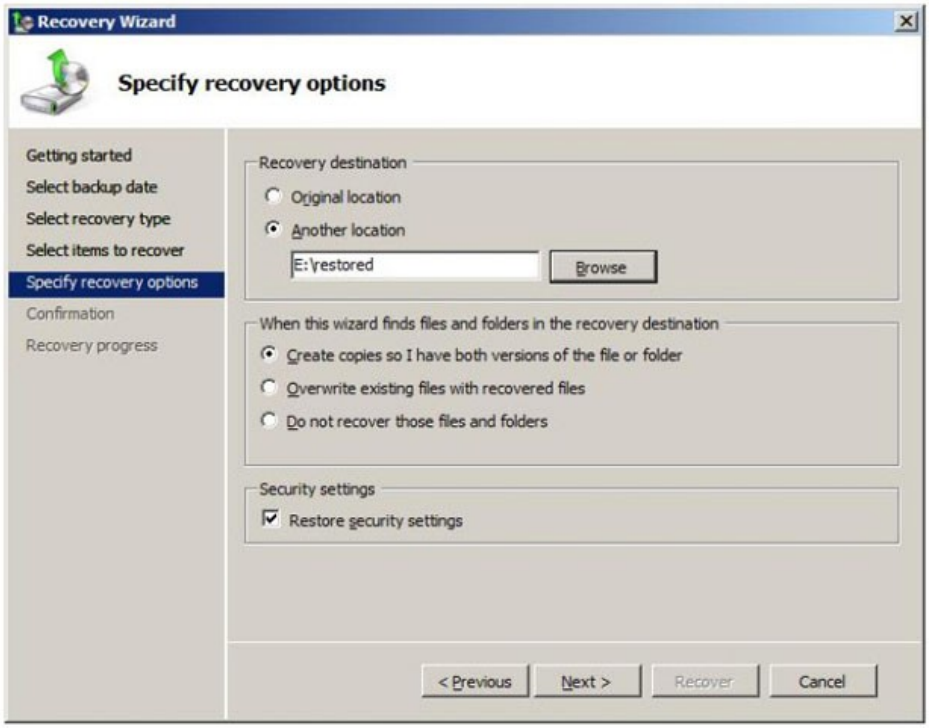


Рис. 9.8. Параметры восстановления

Задание

Выберите из архива, созданного в предыдущей части работы, группу файлов для восстановления. Восстановите их в первый раз *по* исходному пути с сохранением копий, во второй раз - *по* альтернативному пути. Опишите, в чем разница в полученных результатах.

Теперь рассмотрим организацию резервного копирования *по* расписанию. Для этого в **Windows Server Backup** выберем опцию **Backup Schedule**. Первое окно запустившегося

мастера информирует, что прежде чем устанавливать *резервное копирование по* расписанию, нужно определить:

- что будет копироваться (полное резервное копирование сервера или отдельные диски);
- как часто надо проводить копирование;
- где размещать копии.

При этом надо учитывать:

1. даже при выборе резервного копирования отдельных разделов, в их список обязательно должен быть внесен раздел (-ы) с операционной системой;
2. копирование может выполняться один или несколько раз в день;
3. для хранения результатов резервного копирования должен выделяться отдельный диск, внутренний или внешний (например, подключаемый по USB). Перед началом использования, он будет отформатирован мастером архивации. Рекомендуется, чтобы он был не менее, чем в 1,5 раза больше по объему, чем архивируемые диски.

Пусть требуется ежедневно делать *резервное копирование* диска раздела с операционной системой. В окне мастера аналогичном рис. 9.3, выбираем вариант *Custom*, в окне аналогичном рис. 9.4 - *диск* C (на котором расположена *операционная система*). Указываем расписание (рис. 9.9).

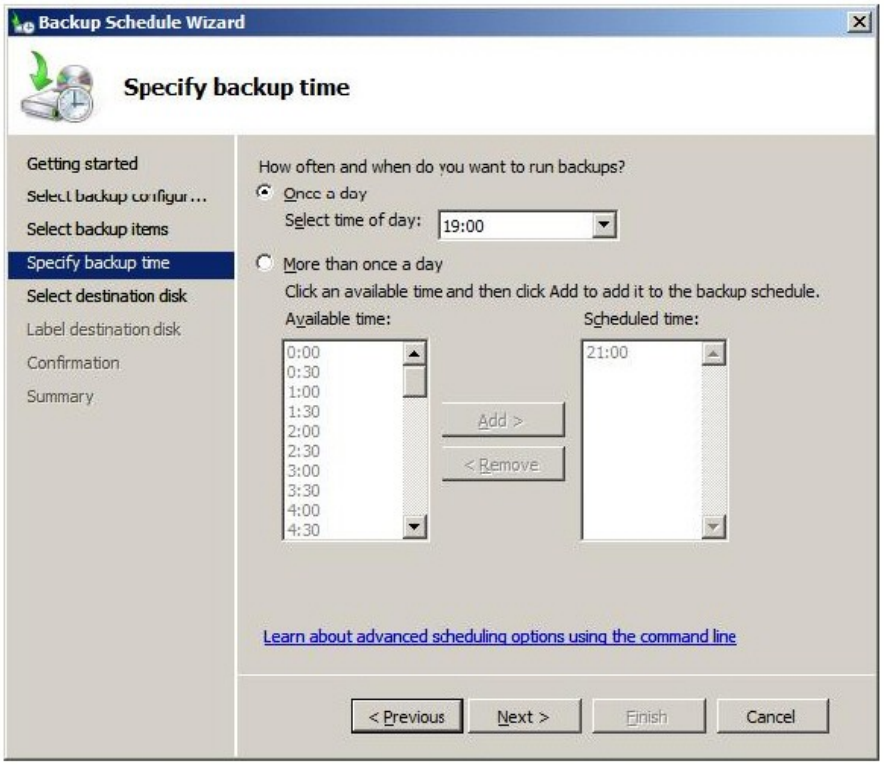


Рис. 9.9. Расписание резервного копирования

Дальше определяется *диск* (рис. 9.10), он может быть не отформатирован. Диску будет назначена *метка* с названием сервера и датой определения резервного копирования, после чего будет проведено *форматирование*. Диску не назначается буква и он не будет доступен пользователям как обычный *диск*.

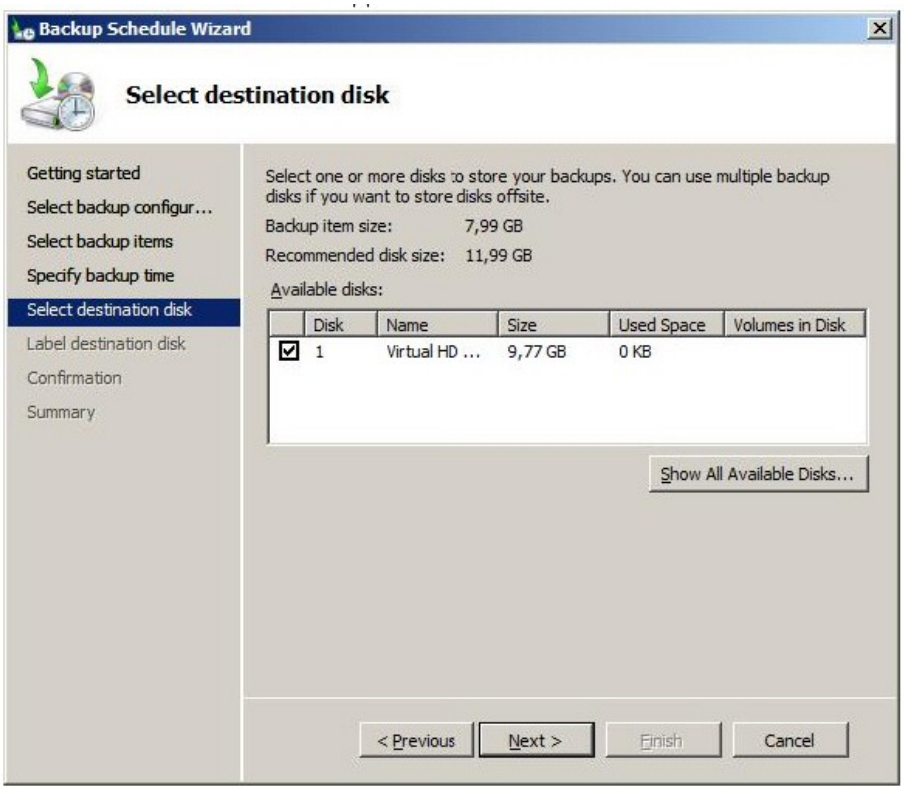


Рис. 9.10. Диск для хранения резервных копий

Когда работа *по* настройке автоматической архивации завершена, можно сделать дополнительные настройки, повышающие *быстродействие* для отдельных дисков. Для этого в списке **Actions** в оснастке **Windows Server Backup** выберите *пункт* **Configure Performance Settings**. В открывшемся окне (рис. 10.11) можно установить, какой тип резервного копирования производить для диска – полное (**full**) или добавочное (**Incremental**). *По* умолчанию используется полное. Добавочное помещает в *архив* только измененные с момента последнего архивирования файлы, это позволяет провести *резервное копирование* быстрее, но более существенно снижает *производительность* сервера в период копирования (т.к. надо проводить проверку).

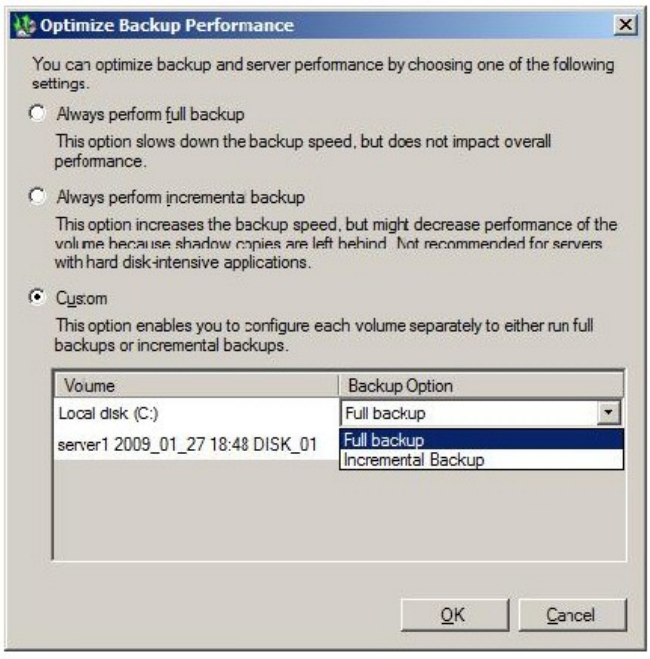


Рис. 9.11. Выбор типа резервного копирования для диска

Порядок восстановления такой же, как и при однократном копировании. Кстати, посмотреть параметры запланированного резервного копирования можно с помощью оснастки **Task Scheduler** (рис. 10.12).

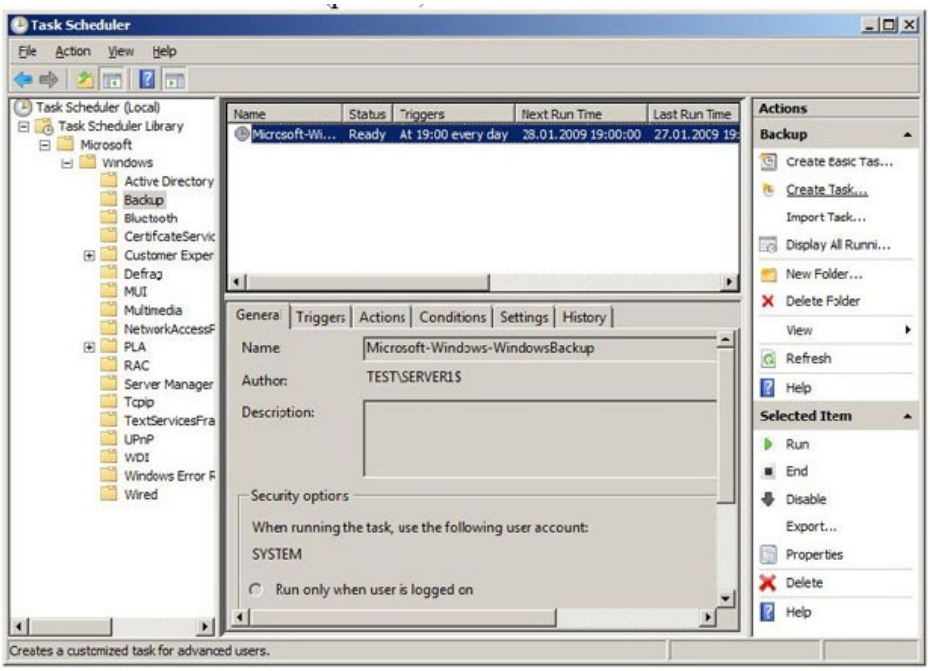


Рис. 9.12. Параметры созданного задания

Задание

Разработайте и реализуйте план ежедневного резервного копирования раздела с операционной системой.

При выполнении лабораторной работы на виртуальной машине для хранения резервных копий можно подключить дополнительный виртуальный *диск* (настройка делается в свойствах виртуальной машины, когда она не запущена). При выполнении работы на учебном севере, заранее определите физический *диск*, на который можно сохранить копии (*диск* не должен содержать полезных данных, т.к. он будет отформатирован!).

Выберите такое время создания копии, чтобы результат можно было увидеть в ходе выполнения лабораторной.

После создания копии, восстановите какой-либо из файлов.

Используя опцию **Backup Schedule** оснастки **Windows Server Backup**, удалите запланированное задание на *резервное копирование*.

Контрольные вопросы:

1. Утилиты администрирования.
2. Выбор дисков для резервного копирования.
3. Доступные резервные копии для выбранного сервера.
4. Выбор типа и параметров восстановления.
5. Диск для хранения резервных копий.
6. Выбор типа резервного копирования для диска.

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-3	1-2	1-3

Оценочные средства: собеседование.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

3.1. Рекомендуемая литература

3.1.1. Основная литература:

1. Царев, Р.Ю. Программные и аппаратные средства информатики: учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670
2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

3.1.2. Дополнительная литература:

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412
2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

1.3. Методическая литература:

1. Методические указания по выполнению лабораторных работ по дисциплине «Программно-аппаратные средства защиты информации».
2. Методические указания по выполнению практических работ по дисциплине «Программно-аппаратные средства защиты информации».
3. Методические рекомендации для студентов по организации самостоятельной работы по дисциплине «Программно-аппаратные средства защиты информации»

3.1.4. Интернет-ресурсы:

http://www.biblioclub.ru/ - электронная библиотека
http://www.uts-edu.ru/ - «Электронные курсы»

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

для обучающихся по организации и проведению самостоятельной работы по дисциплине «ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ»

для студентов направления подготовки **10.03.01 Информационная безопасность**

направленность (профиль) **Безопасность компьютерных систем**

Пятигорск, 2024

СОДЕРЖАНИЕ

1. Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

− формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

− написание докладов;

− подготовка к семинарам, практическим и лабораторным работам, их оформление;

− составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);

− выполнение учебно-исследовательских работ, проектная деятельность;

− подготовка практических разработок и рекомендаций по решению проблемной ситуации;

− выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;

− компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;

− выполнение курсовых работ (проектов) в рамках дисциплин;

− выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

● подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);

● основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);

● заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине «Программно-аппаратные средства защиты информации» направлена на формирование следующих **компетенций**:

Код	Формулировка:
ОПК-6	Способен при решении профессиональной задач организовывать защиту информации по ограниченного доступа в соответствии с нормативными правовыми актами, нормативными и методическими документами Федеральной службы безопасности Российской Федерации, Федерации службы по техническому и экспортному контролю.
ОПК-1.4	Способен оценивать уровень безопасности компьютерных систем и сетей, в том числе в соответствии с нормативными и корпоративными требованиями

2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

● систематизация и закрепление полученных теоретических знаний и практических умений студентов;

● углубление и расширение теоретических знаний;

● формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;

● развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

● формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

● развитие исследовательских умений;

● использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

3. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
5 семестр					
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Самостоятельное изучение литературы и источников	Собеседование	12,96	1,44	14,4
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к лабораторным занятиям	Отчет письменный	9,72	1,08	10,8
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к практическим занятиям	Отчет письменный	6,48	0,72	7,2
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к лекциям	Собеседование	3,24	0,36	3,6
		Итого за 5 семестр	32,4	3,6	36
6 семестр					
ОПК-6	Самостоятельное	Собеседован	26,64	2,96	29,6

(ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	изучение литературы и источников	ие			
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к лабораторным занятиям	Отчет письменный	8,64	0,96	9,6
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к практическим занятиям	Отчет письменный	2,88	0,32	3,2
ОПК-6 (ИД-1 ИД-2 ИД-3) ОПК-1.4 (ИД-1 ИД-2 ИД-3)	Подготовка к лекциям	Собеседование	1,44	0,16	1,6
Итого за 6 семестр			39,6	4,4	44
Итого			72	8	80

4. Порядок выполнения самостоятельной работы студентом

4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;

2. Выделите главное, составьте план;

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

4.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

4.4. Методические рекомендации по написанию научных текстов (докладов, докладовов, эссе, научных статей и т.д.)

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т, п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад - это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки

а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы - изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

Требования к написанию доклада

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

Структура доклада:

● Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

● Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

● Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

● Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

● Приложение (при необходимости).

Требования к оформлению:

● текст с одной стороны листа;

● шрифт Times New Roman;

● кегль шрифта 14;

● межстрочное расстояние 1,5;

● поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;

● доклад должен быть представлен в сброшюрованном виде.

Порядок защиты доклада:

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

Оценка доклада

Доклад оценивается по следующим критериям:

• соблюдение требований к его оформлению;

• необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;

• умение студента свободно излагать основные идеи, отраженные в докладе;

• способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

Критерии оценки:

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

Оценка «хорошо» выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

Оценка «удовлетворительно» выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

Оценка «неудовлетворительно» выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

4.5. Методические рекомендации по выполнению исследовательских проектов

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на

основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласоваться с преподавателем.

Требования по выполнению и оформлению проекта

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

Критерии оценки работы участника проекта.

Для каждого из участников проекта оцениваются:

• профессиональные теоретические знания в соответствующей области;

• умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;

• умение работать с техническими средствами;

• умение пользоваться соответствующими выполняемому проекту информационными технологиями;

• умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;

• умение работать в команде;

• умение публично представлять результаты собственной деятельности;

• коммуникабельность, инициативность, творческие способности.

Критерии выставления оценки участникам проекта

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
«Отлично»	Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен,	Технические средства и информационные технологии освоены и использованы для реализации	Студент проявил инициативу, творческий подход, способность к выполнению сложных	Проект представлен полностью и в срок.

Оценка	Профессиональные компетенции	Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий	Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)	Отчетность
	допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом.	проекта полностью	заданий, навыки работы в коллективе, организационные способности.	
«Хорошо»	Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.	Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий	Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.	Проект представлен достаточно полно и в срок, но с некоторыми недоработками.
«Удовлетворительно»	Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.	Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующими информационными технологиями	Студент выполнил большую часть возложенной на него работы.	Проект сдан со значительным опозданием (более недели) и не полностью
«Неудовлетворительно»	Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта.	Навыков работы с техническими средствами нет, информационные технологии не освоены	Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте.	Проект не сдан.

Студенты должны: защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (*если проект групповой*).

Максимально возможный балл за весь текущий контроль устанавливается равным **55.** Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

4.6. Методические рекомендации по подготовке к экзаменам и зачетам

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неутомительные занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Контроль самостоятельной работы студентов

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

Список литературы для выполнения СРС

Основная литература:

1. Царев, Р.Ю. Программные и аппаратные средства информатики: учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература:

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Методическая литература:

1. Методические указания по выполнению лабораторных работ по дисциплине «Программно-аппаратные средства защиты информации».
2. Методические указания по выполнению практических работ по дисциплине «Программно-аппаратные средства защиты информации».
3. Методические рекомендации для студентов по организации самостоятельной работы по дисциплине «Программно-аппаратные средства защиты информации»

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2. http://www.uts-edu.ru/ - «Электронные курсы»